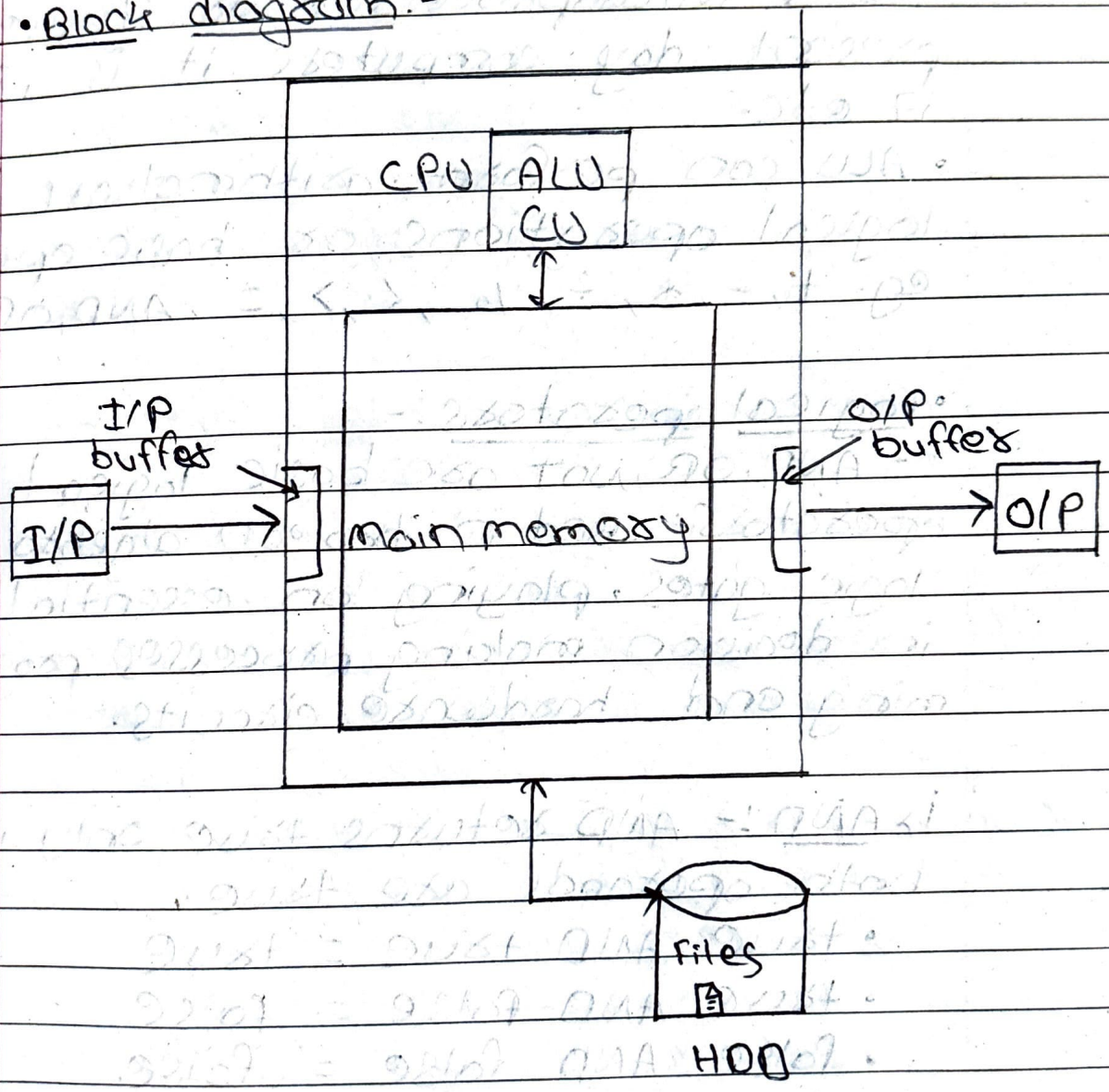
C++ Programming

* Essential Fundamentals:-

2. How do computers work:-

• Block diagram:-



• CPU is central processing unit.
• It has two parts ALU (Asithmatical logical unit) and CU (control unit).
• HDD is the hard disk drive which consists the data in form of files.

- I/P is input device like mouse, keyboard etc.
- O/P is output device such as monitor
- CPU is the heart of the computer or a microprocessor or if we see present day computers it is i3, i5, i7 etc.
- ALU can perform arithmetical and logical operations or basic operations eg. +, -, *, ÷, %, <, >, =, AND, OR, NOT

- Logical operators :-
  - AND, OR, NOT are basic logical operators used in boolean algebra and logic gates, playing an essential role in decision making processes, programming and hardware circuits.

i> AND :- AND returns true only if both operands are true.
- true AND true = true
- true AND false = false
- false AND false = false.

ii> OR :- OR returns true if atleast one of the operands is true.
- true OR true = true
- true OR false = true
- false OR false = false.

iii) NOT:- NOT is a unary operator (acts on a single operand) that inverts the boolean value.

- NOT true = false
- NOT false = true.

-In programming languages like C++, Java or python, you might encounter these operators as &&, ||, ! respectively.

- Boolean algebra:- Boolean algebra is a branch of mathematics that deals with binary values (0 and 1), which correspond to false and true in logic.
  -It is used to perform logical operations on binary variables.

- Logic gates :- Logic gates are physical devices used in digital electronics that implement Boolean algebra operations -They are building blocks of digital circuits, helping process binary information.

- CU :- Computers has many parts connected such as input device, output devices, HDD etc, control unit is used to utilize all the resources.

- HDD :- contains files that may be of two types as follows:-
  i) Program files - eg. Notepad file.
  ii) Data files - eg. my.txt file (generated by notepad) -, video files, audio files etc.
  - HDD is permanent memory.

- Main memory :- If we want to execute the program, then it should be present in main memory so that CPU could access it.

- Input buffer :- An input buffer is a temporary storage area where data is held before it is processed by a program.
  - when data is received from an input device, it is first stored in the input buffer to ensure the system doesn't miss or loose any data if the processor isn't ready to process it immediately.

- Output buffer :- Place where data is held before it is sent to an output device.
  - It allows the system to store data temporarily until the output device is ready to handle it.

Date
Page

- Computers work on binary numbers i.e 0,1. 0 is low volt and 1 is high volt or 0 is false and 1 is true.

- Number System:-

- Binary = {0,1}
- octal = {0,1,2,3,4,5,6,7}
- Decimal = {0,1,2,3,4,5,6,7,8,9}
- Hexadecimal = {0,1,2,4,5,6,7,9,9,A,B,C, D,E,F}

* Binary ↔ Decimal conversion.

$25_{(10)} = 11001_{(2)}$

2	25	
2	12	-1 ↑
2	6	-0
2	3	-0
2	1	-1
	0	-1

$11001_{(2)} =$

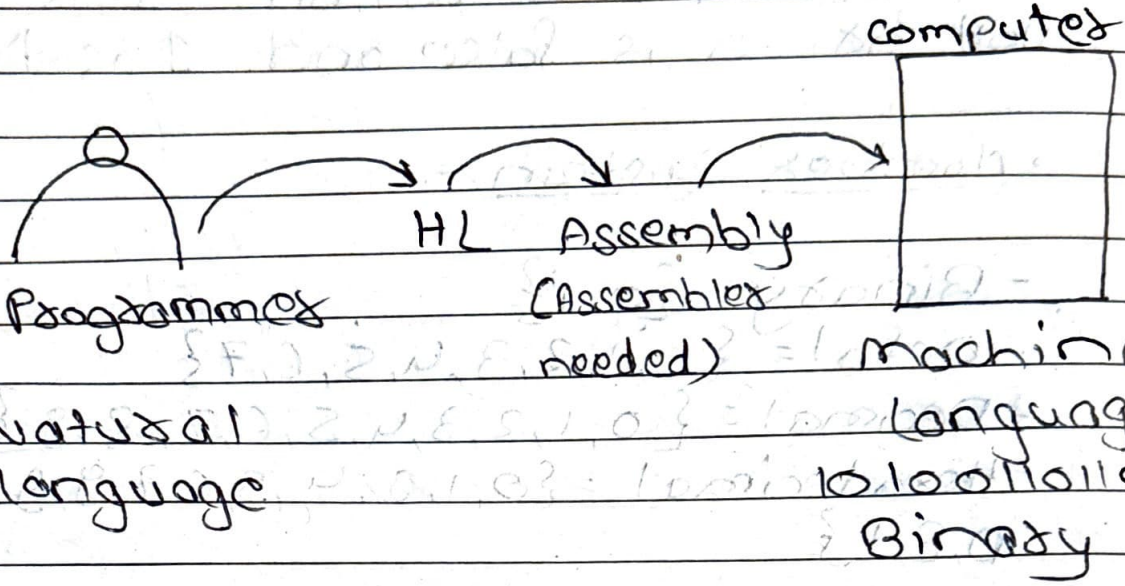
1	1	0	0	1
2^4	2^3	2^2	2^1	2^0

$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 16 + 8 + 0 + 0 + 1$

$= 25$

* Low-level vs High level language:-



Low-level languages:- closed to machine
• machine understandable:
- Assembly language
- machine language.

High-level languages:- • closed to programmer
• should be converted into ML in order to machine to understand
- C
- C++
- Java
- python
etc.

• High level languages can be categorized into compiler based, interpreted based or hybrid.

* Compiled vs interpreted :-

Tasks for compiler or interpreter are:-

1) check errors.
2) Convert into machine code.
3) Execution of program.

Compiler :- Compiler is a program or software
- Compiler doesn't take the responsibility of execution but interpreter does.
- Compiler generates a .exe files, which is a executable file.
- The .exe file is a machine code file.
- If there is any error in the source code, the program will not get compiled.
-

Interpreter :-
- Interpreter translates the code line by line in machine language and executes it.
- If there is any error, then too file will be translated and executed.
- Compiled program runs independently, but interpreted program runs inside interpreter. (dependent)

- Compiled programs are fastest because they are independent.
eg. chinese recipe.

Hybrid :-

Java
↓ compiler
byte code
↓ JVM
machine code

* What is operating system:-

- OS is a master program which uses all the resources of computer system and provides services to the user.
- operating system is a program or a set of programs which loads up when a computer starts.
- The When a computer starts, the OS is put into the main memory and ~~comp~~ CPU starts executing it.
- When we run a program, the OS will bring that program in main memory, and tells CPU to execute it.
- If a program wants to show something on monitor, it cannot directly do it.
- the OS does it for program.

Program
↓ System call.
O.S

* Programming Paradigm:-

- Style or technique of programming.

1. Monolithic programming :-

- In this, entire program code will be single piece or single file. or single body. eg. COBOL (Common Business oriented lang.)
- one programmer

2. Modular / Procedural programming :-

- use of functions
- Reusing of code (no need to write it twice)
- Dividing into smaller tasks (functions).
- multiple programmers.
  eg. C language.
- functions are used to performing tasks.
- These tasks are performed on data.
- In C language we use structures to define data.

3. Object oriented :-

- Intro to classes.
- Contains data and functions upon that data.
- i.e data members and member functions.
- eg. c++, java.

- Everything the logic, etc is same, the way we organize is different.

* Algorithm and Pseudo code :-

An algorithm is the abstract idea or logic behind solving a problem, while pseudo code is the way of expressing that algorithm in a more structured, code-like format without strict syntax rules.

* Steps for program development and execution :-

1) Editing :-

- Editing means writing a program in IDE or in a environment.

2) Compiling :-

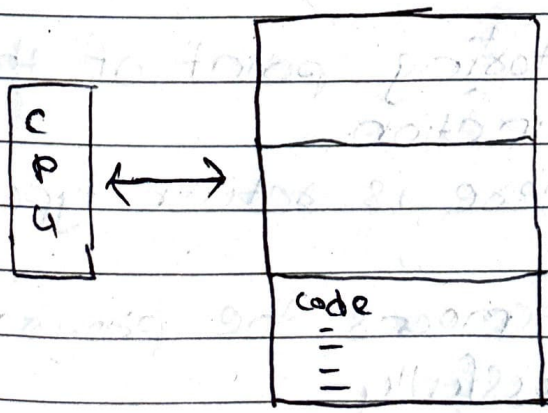
- Compiler will compile the source code into machine code.
- first.cpp → first.exe [executable file]

3) Linking library :-

- Language provides some ready made code which makes easy for us to write a program.
- #include <iostream> is a header file which is supported by a library file.
- So when you are using the library means you are copying code of library into your exe file.
- If we use header file, then code of it is copied in .exe file.

Loading

4) Main memory is divided into 3 sections



- Getting the program from hard disk into the main memory for execution by the CPU. This process is called loading.
- All this is done by OS.

5) Execution :-

- For variables, space is allocated inside stack.
- Heap is used for dynamic memory allocation.