

CS-101 (LECTURE 1)

Data:-

All kinds of information including pictures, letters, numbers and sounds.

- Modern computers have a defining feature that distinguishes them from older ones that they can be programmed.

Program:- a list of instructions given to a computer.

History:- The word computer was first recorded in 1613, referring to a person who carried out calculations or computations.

- The word continued to be used in that sense until the middle of 20th century, (before the development of modern computers.)

Computer History

↓ ↓
Automated Calculations Programmability.

- Early computers were mechanical calculation devices e.g. Abacus 3000 BC.

ENIAC ⇒ Electronic numerical integrator and computer.

INFORMATION SYSTEM

It has the following five parts.

- (i) People (end users).
- (ii) Procedures (specific rules/guidelines for computers).
- (iii) Software (provide step-by-step instructions).
- (iv) Hardware (includes peripherals).
- (v) Data. (consist of unprocessed facts, texts, no.s & images).

Most imp part of any system is "People".
Their contact is either

↓
Direct

↓
indirect.

SOFTWARE

SYSTEMS

A collection of programs that enables application software to interact with the hardware.

A background software that helps the comp. in managing its resources.

APPLICATIONS

An end user software that serves two categories.

- (i) General purposes
- (ii) Special purposes.

Qno. 1 (b) Assignment.

TYPES OF COMPUTERS

(i) SUPER COMPUTER.

(EXAMPLE:- BELEDEEP BLUE.)
etc
Used by very large organization.
High capacity
Tracking space & weather.

(ii) MAIN FRAME:-

Not as powerful as super computers.
occupies air conditioned rooms & wired.
Capable of great processing & calculations.

(iii) MINI COMPUTERS:- (E.G:- Mitra 1115, CDC1700)

Midrange computers
Used by mid-size companies or departments
of large companies.

(iv) MICRO COMPUTER:-

least powerful.
widely used
Personal P.C.

(EXAMPLE :- HP, DELL Laptops).

→ Laptops

→ Tablets

→ Desktops

→ Handheld

Hardware

(i) System unit

(ii) Input/Output

devices

(iii) S. storage

(iv) communication

INFORMATION

Processed form of raw data is called information.

Both information and data are stored in electronically.
Document files
worksheet
Database

in files).

Presentations.

Database

* largest connectivity network is -the internet.

'DOCUMENT FILES'

Created by word processors to create & save documents such as memos, term papers & letters.

'WORKSHEET FILES'

Created by electronic spreadsheets to analyze things like budget and predict sales.

'DATABASE FILES'

Created by database management programs to contain highly structured and organized data.

(Example: speaker notes, electronic slides)

"PRESENTATION FILES".

Created by presentation graphics organizations to contain presentation material.

CS - 101 (LECTURE NO. 2)

(No. 2 (a) Assignment.
PARTS OF COMPUTER.

HARDWARE.

All the visible components

SOFTWARE

The parts which we cannot see.

Date: 20
MTWTFSS

PCI slot \Rightarrow network modem AGP slot \Rightarrow graphic card slot MTWTFSS

integrated drive electronics serial advanced technology ...

MOTHER BOARD

- It is also known as PCB (Central Printed Circuit Board).
- where the components reside (Board.)
- It provides slots to host microprocessors, memory, driver controllers, graphics card etc.
- Also provides hosted connections for communication.

HHD:-

- Permanent storage device.
- stores data, programs and OS installations.

HDMI:- (High definition multimedia interface)

- transferring of compressed video data and compressed/uncompressed digital audio data.

* MOST electronic components of a computer are integrated Circuits.

IDE/ATA:-
harddisk / CD ROM
Connector.
Component of MB.

COMPUTER CHIPS:-

(i) DIP

(Dual inline
Package)

two row of pins that connect IC circuitry to a circuit board.

(ii) DIMM

(Dual in line
Memory Module)

a small circuit board containing chips to store memory.

(iii) PGA

(Pin Grid
Array)

single edge contact

cartridge used in was microprocessors pioneered

square chips

microprocessor

Pentium 3 by intel

[Memory device] → [RAM] → [CACHE] → [CPU].

CPU can access RAM better than storage device Date: 20
keep up with CPU so cache is used.

RAM:-

- A temporary holding area of data application program instructions and the OS.
- RAM is Primary or main storage memory.
- It is measured in GB or MB.

TYPES OF RAMS:-

They vary in speed, technology and configuration.

- (i) SDRAM
- (ii) RDRAM
- (iii) DDR
- (iv) DDR2
- (v) DDR3

(Synchronous Dynamic RAM)
(Rambus Dynamic RAM)

- The speed is measured in nano seconds.
 $1\text{ns} = 10^{-9}\text{s}$. (1 billionth of a second).
- It can also be expressed in MHz.

ROM. (Read only memory)
(Permanent & non-volatile memory).
(It holds the computer's startup routine).
Only way to change the inst of ROM is to change the chip.

CMOS :- (complementary metal oxide semiconductor).

- Retains the data even after shutdown but requires a minute amount of energy.

STORAGE DEVICE & MEDIUM.

Storage medium is the part of storage device that retains / contains data & "Storage" can record & retrieve data from that medium.

FOLLOWING DEVICES ARE IMP:-

- (i) Optical storage
- (ii) Magnetic storage (HDD, Magnetic tape, Floppy)
- (iii) Solid state storage.

HARD DRIVE:-

- DISK coated with Magnetic iron oxide. (hard disk platter).
- (Hard disk) one or more platters with a read write head.

HEAD CRASH:- Phenomena of Hard drive.

- Read write head exposes to dust and cause data loss (may be caused by jarring).

OPTICAL STORAGE:-

data stored in light spot (lands) & dark spots called as (pits), more safer than magnetic media.
e.g. CD 700 MB.
DVD 4.7 GB (2x 8.5 GB)

Blueray 50 GB

SOLID STATE STORAGE

- stores data in non-volatile, erasable, low-power chip.
- Some SSD require a "card reader" to transfer data to or from a computer.

→ faster & portable.

"MOORE'S LAW"

Since the invention of IC in 1958, the no. of transistors that can be placed inexpensively on an IC has increased exponentially, doubling every 2 years.

"Moore's law is an observation that the no. of transistors in a dense integrated circuit doubles approximately every 2 years".

(Qno.3(a)) Assignment.

DATA REPRESENTATION :-

DIGITAL DEVICES :-

- A digital device that works with discrete data 1 & 0.

IMPORTANCE :-

- Most Modern Computers do not use decimals to represent data either Binary numbering system (base2)

→ Consisting of 2 digits 1 & 0

→ uses powers of 2 rather than 10.

Date: 20
MTWTFSS

(USB & compact flash).

DECIMAL (Base-10)
0 1 2 3 4 5 6 7 8 9

BINARY (Base-2)
0 1 10 11 100 110 111 1000 1001 1010 1011 10100 10101 10110 10111 11000 11001 11010 11011 11100 11101 11110 11111 10000

CONVERSIONS (IMP)

BINARY TO DECIMAL

$$(1011)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ = (11)_{10}.$$

DECIMAL TO BINARY

$$\begin{array}{r} 2 | 125 \\ 2 | 62 \\ 2 | 31 \\ 2 | 15 \\ 2 | 7 \\ 2 | 3 \\ 2 | 1 \end{array} \quad \begin{array}{r} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{array} \rightarrow (125)_{10} = (111101)_2$$

Computer digitizes every thing into 1s & 0s.

:-

one binary digit
0 or 1

BYTES :-

8 bits.

* Bit become electrical pulses that travel over the circuit.

LECTURE NO. 3.

(Qno.3(c))

"COMPUTER SOFTWARE"

Consists of computer programs, data files that work together to provide instructions & data of how to carry out certain tasks.

SYSTEM SOFTWARE

designed to help comp. carry out basic operations like OS utilities, device drivers.

APPLICATION SOFTWARE :-

designed to be used for a specific purpose.

Eg:- education, medical etc.

[Hardware] → [System software] → [application software]
Windows.

OPERATING SYSTEM:-

The master controller of all the activities that take place inside a computer.

- It manages system resources e.g.-
 - (i) Processor
 - (ii) Memory management → no data leakage or corruption.
 - (iii) Storage resources.
- Ensures input & output manages/works properly.
- Establish basic elements of user interface.
- Remembers locations and names of the files.
- Hides details of operation of hardware from the applications. → it simplifies applications remove dependence on hardware.

EXAMPLE:- Nls.Nord → OS → Drivers → Printer.

if we want to print something, OS can link up to command the drivers and communicate document data.

* Small Os is stored in the ROM & larger are stored in HHD.

HOW DOES THE OS START WORKING?

- Switching on a computer initiates a series of POSTS (Power ON self test). The BIOS is in control at this point. **BIOS** = UEFI (unified extensible firmware interface), that runs a series of diagnostic test to check hardware components like CPU, RAM
- Next, the BIOS loads a very small program in the very first location of hard disk, placed into RAM and processor starts running it. This is called "Boot loader / **Bootstrap** program".
Bootstrap then allows
- ↑ Now the **OS** from ROM takes control to provide a userfriendly environment to run application.
- Now as the OS is loaded but initially the windows' **Kernel** is loaded.
→ "The core part of OS responsible for management of hardware resources, processes, memory and other essential functions." i.e imp. OS services.

FIRMWARE:-

SMALL PROGRAMS CONTROLLING ELECTRONIC DEVICES.

- typically reside ROM. (startups & low level I/O routines)
- more complex resides in flash memory to allow updates.

FIRWARE is in the smartphone's components.

Date: 20
MTWTFSS

LECTURE NO. 4

(Q no.3(b)) Assignment. x

ARTIFICIAL INTELLIGENCE:-

A program that can sense, adapt, reason & act (intel)

PROGRAMMING:- Step by step instructions given to a computer to run a program:

AI :-

Developing systems that can perform (human like intelligent) task

→ AI learns from patterns in data to make informed decision. Its sub fields are

- (i) Machine learning
- (ii) Neural Networks.
- (iii) Natural Language Programming.

INTELLIGENT SYSTEMS:-

- (i) Stimulus translated into internal messages (like nerves detection).
- (ii) the representation is manipulated in the cognitive system (i.e Brain)
- (iii) the action is generation. (nerves respond)

APPLICATION & SIGNIFICANCE:-

Healthcare:-

AI can analyze data and help in diagnostics, personalize treatment etc.

Autonomous Vehicles:-

AI enables self driving cars to navigate safely (etc)...

COMPUTER VS. BRAIN

COMPUTATIONAL UNITS.

1 CPU, 10^3 gate

10^{11} neurons

storage units.

10^{12} bit disk, 10^{11} bits RAM

10^{11} neurons, 10^{14} synapses

10^{-9} sec.

10^{-3} sec.

Cycle time.

10^{10} bits/sec

10^{14} bits/sec.

10^9

Memory update.

10^{14}

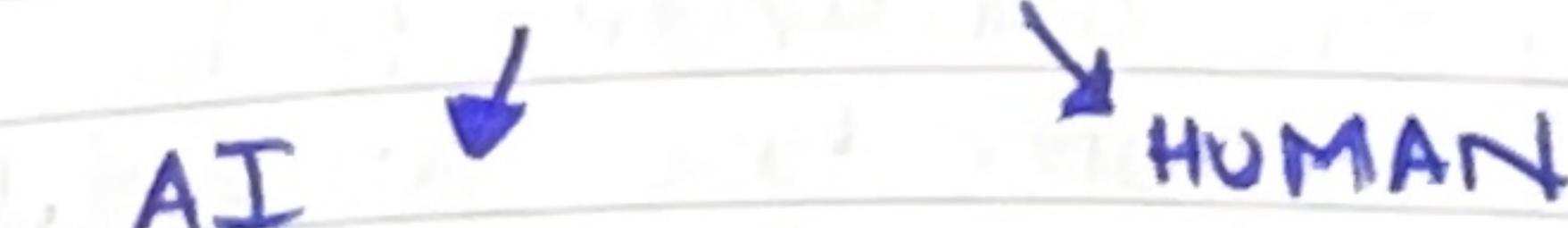
cycle time = production / working time.

Bandwidth = a range of frequencies required or given in a band to transfer a signal.

LEVEL OF ABSTRACTION.

The level in hierarchy of understanding and representing a system.

→ The higher the level, the less details.
→ The lower the level, the higher the details.



(i) Hardware

(ii) Build Brains

(iii) Network

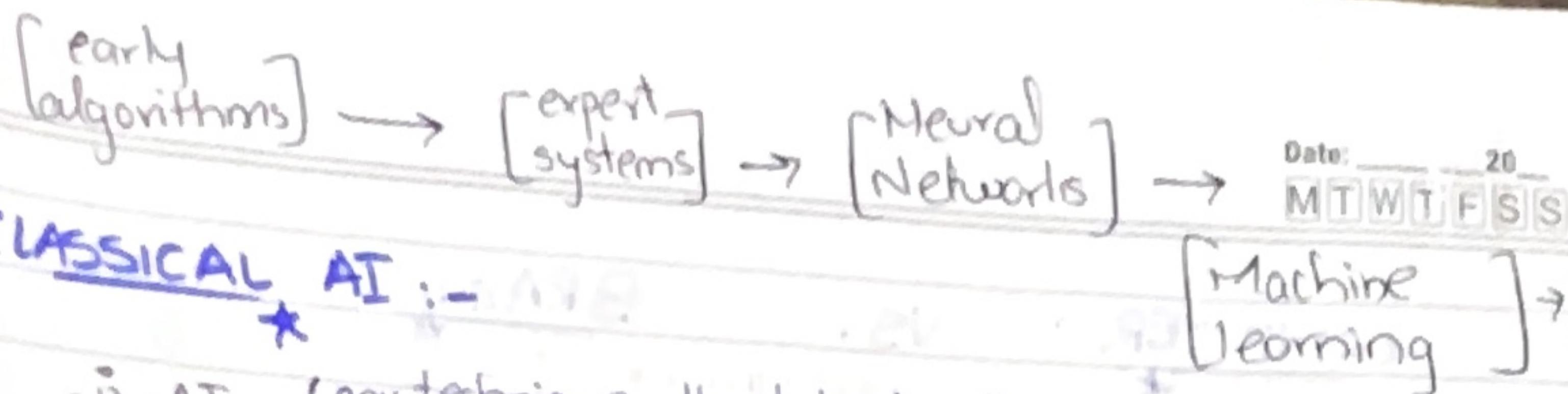
Neural network.

(iv) Algorithm/thinking.

(v) Immediate Behavior

(vi) Task Performance

COMPETENCE



CLASSICAL AI :-

- (i) AI (any technique that helps the comp to mimic human intelligence, using logic, if-then rules)
- (ii) Machine learning (AI that enables machines to improve at tasks with exp.)
- (iii) Deep learning. (subset of M.L. composed of algorithms that allow software to train itself.)

PROBLEMS OF AI :-

- (i) lack of common sense knowledge.
- (ii) doesn't understand Natural language.
- (iii) issues in speech recog. & translation.

TURING TEST :-

The test conducted to evaluate the Computer's intelligence in contrast to a human's intelligence.

EXAMPLE:- Assign same question to a human and computer and the evaluator checks which answer is of a human, if the judge /evaluator fails ... the computer had passed the test

M
Hans

'CS-101 LAB'

C++ code i Task performed: assigning a variable 'a' value and printing it.

```

1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5.     int a;
6.     a=5
7.     cout << a;
8.     return 0;
9. }
10. ~x~
```

} } library is created for writing a code. cin=> command from user cout-> give output

* syntax is very imp to run the code.

CS - LECTURE 4 (CONTINUE).

MACHINE LEARNING:-

It is the study of algorithms for a computer that can:

(i) improve task performance (with experience).

Example:-

A computer is made to play games with itself and learn possible outcomes to eventually increase its winning rate.

MACHINE LEARNING has further 3 aspects

(i) supervised learning (ii) unsupervised learning

(iii) reinforcement learning.

SUPERVISED MACHINE LEARNING:-

In this type of machine learning input objects and the desired output values are used to train a machine model or a computer is trained via "labelled" training data.

Example:-

Data given:- training set $\{(x_i, y_i) \mid i=1 \dots n\}$
Find :- A good approx of $f: X \rightarrow Y$

where $X \notin Y$ are :-

- (i) Spam detection (Maps email to spam/not spam)
- (ii) Digit detection (Maps pixels to $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$)
- (iii) Stock Prediction
- (iv) Specie Prediction.
- (v) House Price Prediction (Regression).

"REGRESSION"

Regression algorithms predict continuous values

"CLASSIFICATION"

Classification algorithms predict discrete values.

It is the process of finding correlation with dependent & independent variables.

It is the process of finding a function which helps in dividing the dataset into classes based on different parameters.

Example

Face, character speech, recognition & Medical diagnosis.

Date: 20
MTWTFSS

UNSUPERVISED LEARNING:-

The type of machine learning that relies on an "unlabelled" data / raw data to train its model.

Unsupervised learning can find underlying patterns and relationships within "unlabelled" data so it's useful for data analysis/exploration, clustering of datasets or projects.

REINFORCEMENT LEARNING:-

The type of Machine learning in which an intelligent agent like a computer interacts with the environments and learn within that through its feed back. (Taking suitable action to maximize reward in particular situation.)

USES:-

- (i) Question Answering
- (ii) Summarization
- (iii) Dialogue generation.
- (iv) Machine translation.

- Game playing,
- Robot in maze.
- Credit assignment problems.
- Multiple agents, observability.

Qno. 2(a) ✓

CPU
Power supply.
GPU.
Sound card.

Keyboard
Mouse
Speaker
Monitor.

Qno. 1 (a)
evolution of computer.

Qno. 2 (a)
✓ SSD vs HHD.

Qno. 2 (b) flowchart.

Qno. 4 (b) P.L(s).

Qno. 5 insight

Qno. 6 T.M.

Qno 2 (a) (Continue assignment).

Date: 20
MTWTFSS

COMPONENTS OF COMPUTER:-

CPU:-

Central processing unit or the CPU is the most important processor in a given computer. Its electronic circuitry executes instructions of a computer program, such as arithmetic, logic, controlling and input/output operations.

This role contrasts with that of external components such as main memory & I/O circuitry and specialized coprocessors such as GPU.

GPU:-

Graphic processing unit is a specialized electronic circuit initially designed to accelerate computer graphics and image processing.

They were also involved in non-graphic calculations involving embarrassingly parallel problems due to their parallel structures, neural networks & cryptocurrency mining.

POWER SUPPLY:-

Electronic device that supplies electric power to an electrical load. The main purpose of P.S. is to convert electric current from a source to correct voltage, current & frequency.

SOUND CARD:-

It is an internal expansion card that provides input /output of audio signals to and from a computer under the control of computer program.

"PERIPHERAL DEVICES"

↓ ↓ ↓ ↓
Keyboard Speakers Mouse Monitor
(input) (output) (input) (output)
text based audio based interaction display/lim
based

Qno. 2 (a) HHD or SSD:-

Hard disk drive is a storage device consisting of magnetic disks to store data whereas SSD uses NAND flash memory to store data similar to the storage in USB drives or memory cards.

I prefer SSD more over HHD.
SSD is better than an HHD;

(i) SPEED:- SSDs offer significantly faster read and write as compared to HDDs.

(ii) DURABILITY & RELIABILITY:-
SSD have no moving parts, they are prone to mechanical failure.

(Ques. (i) (Continue))
(a)

Evaluation of Computer.

(i) 1940 - 1956 (First Generation Computer)
VACUUM TUBES

Vacuum tubes were initially designed by John Ambrose Fleming in 1904. It is used to control flow of electric current in a vacuum. It is used in CRT.

Examples:-

ENIAC
EDVAC
UNIVAC-1.

(ii) 1956 - 1963 (2nd Generation Computers).
TRANSISTORS

These computers used transistors developed in 1947 by three American physicists;

Transistor is a semiconductor used to amplify or switch signals or opens or closes a circuit.

They had high level languages like FORTRAN & COBOL.

Examples:- PDP-8
IBM-1400.

(iii) 1964 - 1971 (3rd Generation Computer)
INTEGRATED CIRCUIT.

ICs were developed in 1958 which is a set of electronic circuits on small flat pieces of semiconductor that is normally known as silicon.

It increased speed and efficiency of the computer. They supported high level languages such as FORTAN-II-V, Pascal, P/L1

Example :- IBM 360/3 series
IBM 36500.

(iv) 1971 - PRESENT (4th generation)
MICRO PROCESSORS!

The micro processors were invented in the 1970s, and the first is called intel 4004.

It contains circuits required to perform, arithmetic, logic, and control functions on a single chip.

Example Apple II.
Atari 8800.

(v) PRESENT & BEYOND (5th generation).

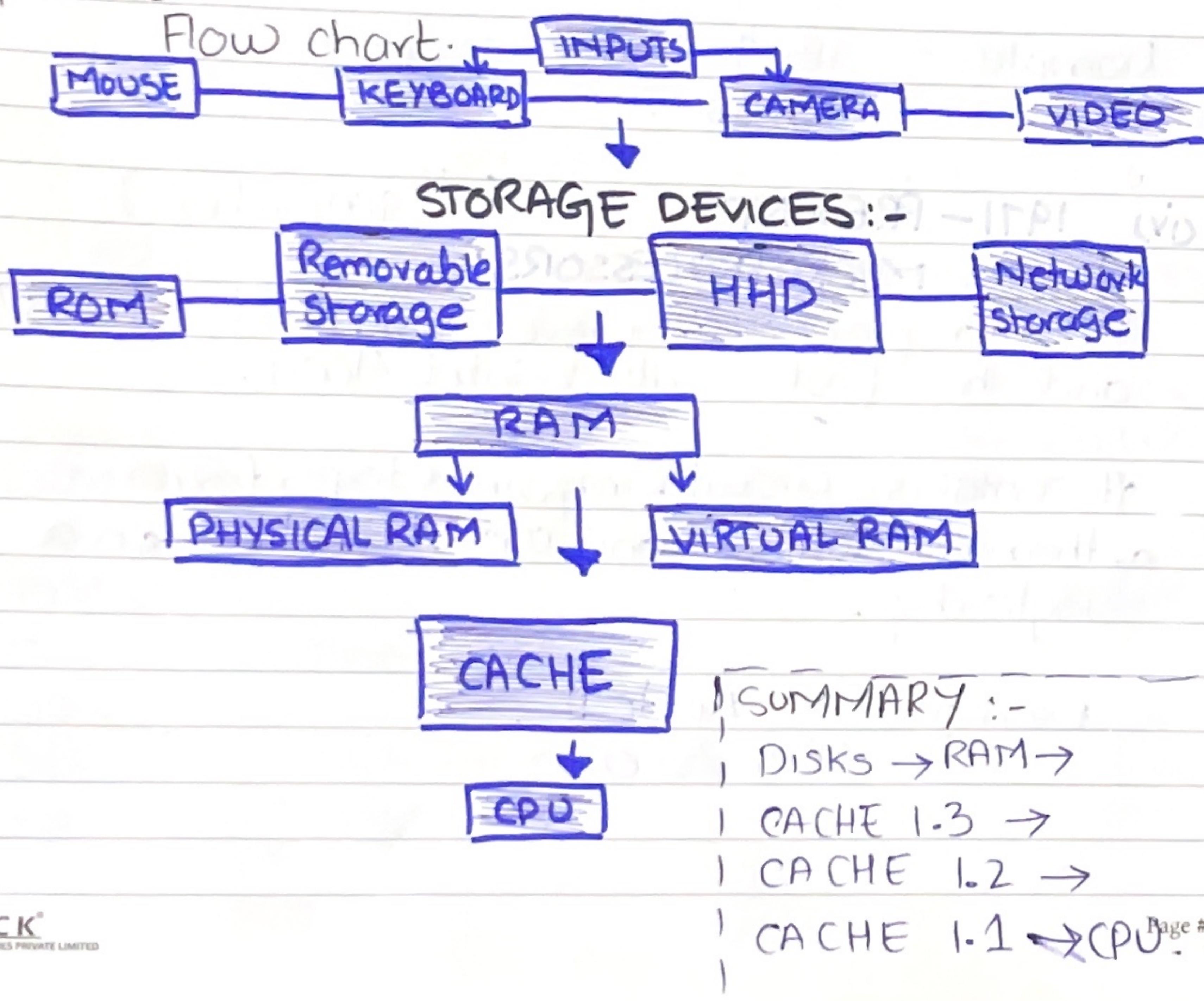
are based upon AI to enhance user interface and task management + performance.

They can operate at almost all high level languages.

Examples:-

notebooks, laptops, desktops.

(Qno. 2 (b)).



Qno. 4 (d)

PROGRAMMING LANGUAGES!

A high level language:- is the one that is user friendly and oriented that has been designed to make it straight forward for a programmer to convert an algorithm into program code.
Example:- Python, C++, C, Java.

LOW-LEVEL LANGUAGE:- are expressed in terms of machine operations that must be performed to carry out a task. This makes writing programs more difficult, as algorithm must be specified in capabilities and specifications of the processor.

They are referred as Machine code or assembly language.

* a high level language is more preferable as it targets a greater audience as compared to low level and thus is more user friendly.

Qno. 5 Reflections & insights:-

The assignment was a good practice of revising, recalling, and learning new concepts older concept

Date: _____ 20
MTWTFSS

eventually making the preparation of assignments much easier.

It cleared many anomalies of mind and thought about AI and gave a much more broad aspect to view the topic.

I personally did not feel any challenge as to appear in the assignment and it seemed a well experience.

Ques.:- Task Manager.

After booting my PC I don't see any program that is using memory.

However if a game were to be opened the memory was being recorded / measured in RAM at 61.0MB.

The task manager itself is using CPU at 2.4% and I wasn't expecting it to be counted at all.

THE END

3rd/4th generation Assembly language
2nd generation Machine language
1st generational Hardware

Date: _____ 20
MTWTFSS

LECTURE NO.5

COMPUTER PROGRAM

CODE.

A program is a step by step inst. that tell a computer how to solve a problem.

The instruction(s) that make up a computer program are known as a code.

PROGRAMMING LANGUAGES!

The code is provided by the programming languages.

Every structure in a programming language has;
Syntax (precise wording)
Semantic (precise meaning)

Programming languages allow machine-to-machine and human to machine interactions.

↓ ↓
LOW LEVEL PROGRAMMING HIGH LEVEL PROGRAMMING

(i) requires inst. for the lowest comp. hardware.

(ii) easy for computer but hard for humans to understand.

(iii) Example:-
Machine & Assembly language.

(i) makes programming for human easier provides commands such as the 'Print' & 'Write'

(ii) Examples:-

- (i) C++
- (ii) Python
- (iii) FORTRAN.

* The high level languages are mostly translated (compiled / interpreted) to machine language in order

MACHINE LANGUAGE / CODE:- to be understood and executed by the microprocessor

The first programming language also referred as 1st generational language.

It consists of a series of instructions in 1s & 0s i.e. the instruction set. Machine language is specific to either a particular CPU or microprocessor.

ASSEMBLY LANGUAGE:-

The 2nd generational language that is the advancement of a machine language.

It consists of abbreviated commands i.e. Mnemonic i.e. ADD, SUB, MUL, DIV, JMP etc.

It is compiled by an assembler however specific to a particular machine as an assembler only compiles the data for a particular computer architecture's assembly language.

COMPILER . (C, C++, C#)

A compiler is a complex software that takes in a program written in a high level language and translates it into some equivalent machine language of a computer. (entirely).

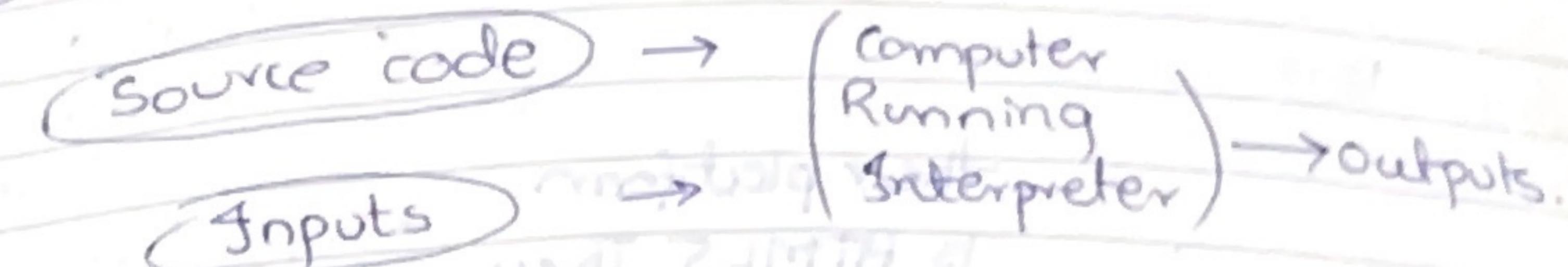
Source code → (compiler) → machine code
Program

Input → (Running Program) → Output
(CPU.)

Between
INTERPRETER:

(Python, Java, JavaScript etc.).

It translates the high level language and executes the source code one-by-one.



IMP QUESTION:-

Difference between a compiler & an interpreter

'COMPILER'

Source code
translates all at
once.

Execution:-

produces an
executable file that
can run independently.

Efficiency .

Faster due to
pre translation.

more challenging as the
code runs entirely and
reported after.

Debugging .

Slower as step by step
translation.

much easier as
analyzed at each step.

Date: 20
MTWTFSS

* Web application
by nature is a website
that is adapted, optimized.
(etc)

Programs:- that can be programmed:-

- (i) Game applications
- (ii) Mobile "
- (iii) Web "
- (iv) Database //

their platform
is HTML5. They
are processed in
a browser built
into the
application.

(i) Javascript.

Building Webs

(ii) Java

Building robust backend servers & android
apps.

(iii) SQL

manage relational databases

(iv) Swift:-

Ios app development.

(v) kotlin:-

App /Android development

(vi) C / C++:-

heavy software (e.g 3D games)

(vii) Python:-

web development, machine learning,
data analysis, automation.

* Native apps of diff Mobs are
programmed differently
(i) iOS → Obj-C DATA 201
(ii) Android → Java DATA 201
(iii) Java → FSS

* Mobile web
apps may look
like apps but
are rather
websites.

* hybrid apps are
downloaded from
play/app store

* native apps are
preinstalled.

* The obj of CS is to solve problems
by developing, analyzing and
implementing
algorithmic solutions.

LECTURE NO. 6:-

ALGORITHM :-

(not written in english as
too rich, imprecise, verbose etc.)
The process of breaking a problem into its
(sub-problems) and solving each problem one
at a time is known as the algorithm.
~ The al-khwarizmi
Principle.

* (in specific order)
* (step by step)

* "Program control" specifies that in which order
statements are they to be executed.

FLOW CHARTS:-

The visual/graphical representation of an
algorithm.

The flow chart employs a series of arrows, blocks
each of which represents a particular operation
i.e step in algorithm.

* The arrows specifically represent the sequence
in which the operation is implemented.



Terminal

Represent start or end.



Flow line

Represent flow of logic.



Process

Represents calculations.



Input/Output

Input & output of info.



Decision

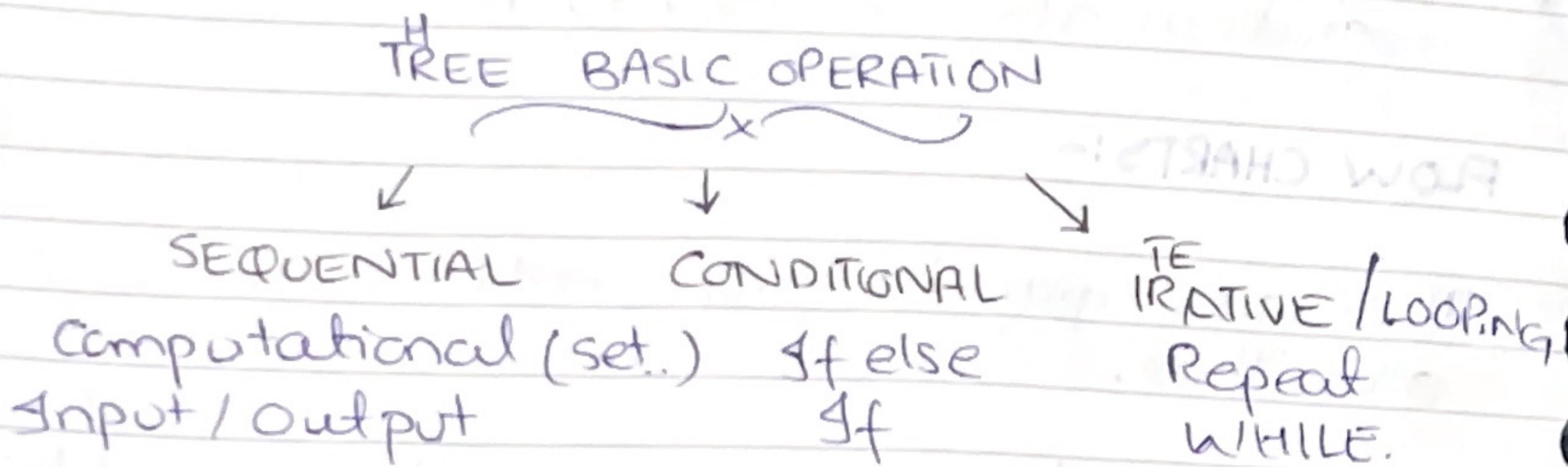
Comparison, question or

FORTRAN: Used in scientific & engineering data
 COBOL: used to manipulate large amount
 PASCAL: Used to teach structured
 Date: 20
 MTWTFSS

PSEUDO CODE :-

Scientists use
 pseudo code to
 express the
 algorithms.

* They are English like statements but modelled to look like programming languages.



LECTURE NO. 7

"History of C++ & C"

C++ has evolved from 'C'

C has evolved from 2 other programming languages

i.e.

- BCPL (Basic Combined Programming language).
- B

* ANSI C is the basis worldwide standards for C programming.

Uses of C++

C++ provides object-oriented programming, i.e. the computer programming model, that organizes software design around data or objects rather than functions or logics.

C++ is object oriented and thus easy to understand.

* C++ programs are built from pieces called 'classes & functions.'

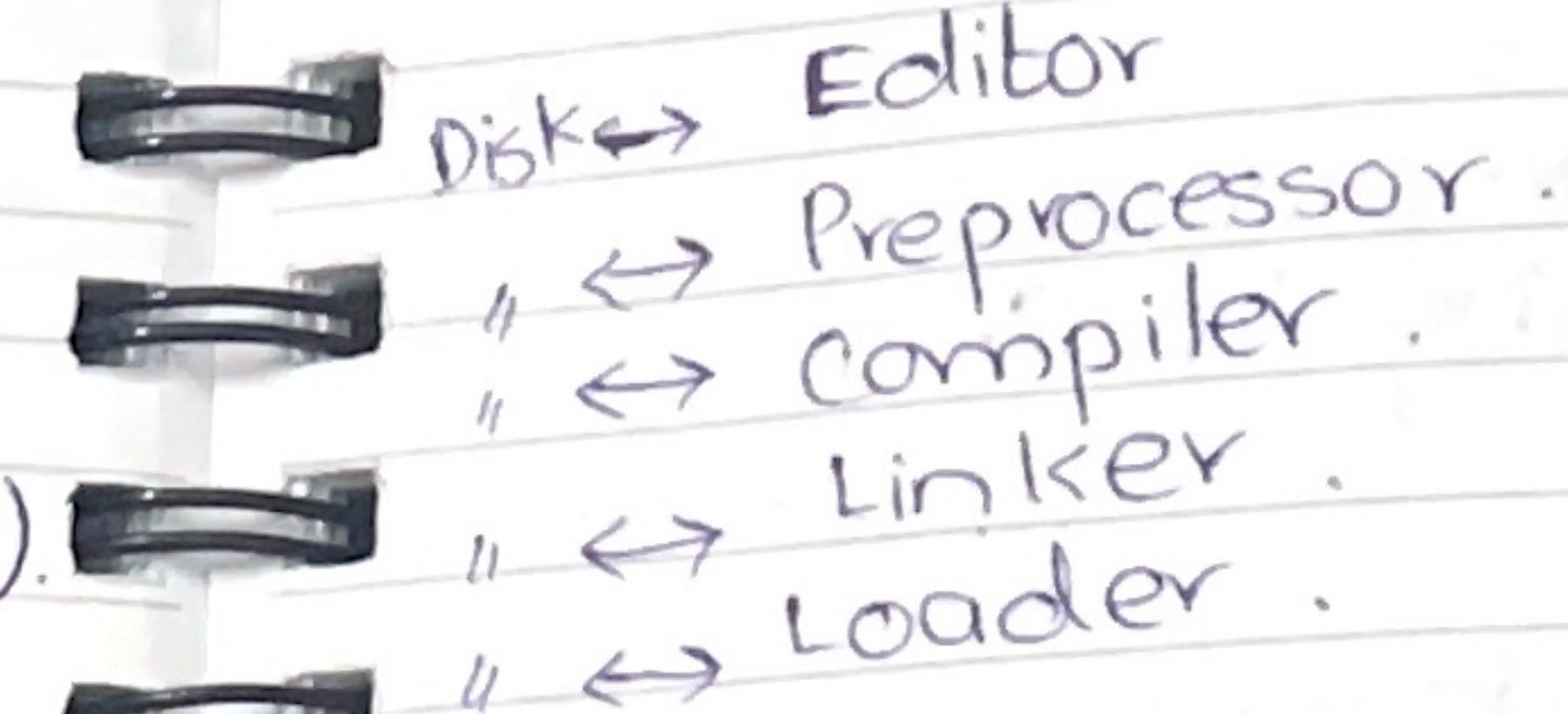
C++ standard library:-

Provides rich collection of existing classes & functions (available for all programmers).

Structured Programming:-

Disciplined approach to writing programs. clear, easy to test & debug, easy to modify.

Stages / components of C++ programming:-



Program is created here & stored ↑
 Preprocessor program process the code
 Compiler creates object & stores it
 links object code with libraries
 puts program on P. memory (creates a.out)



CPU takes each int. and executes it, possibly storing new data values as the program executes.

Editing:-

(i) Editing refers to the typing, writing the program in any text editor, requires all the things in one place like writing the program, compiling and executing it.

(ii) It is achieved by Integrated Development environment (IDE). e.g.: Vs code, X code (GCC), Dev C++.

↳ compiler used for open source C++ compiler that can be used with various editors.

CLANG :-

Another open source C++ compiler that can be used with various editors and IDEs, including Vs code.

C Lion is an IDE that uses clang as its default compiler.

PREPROCESS:-

Before C++ can understand our program it goes through the preprocessing step. This step handles;

- expanding macros (`#define`).
- including header files (`#include`).
- removing comments.

+ each example analyzed one statement at a time.

Date: _____
MTWTFSS
20

It is necessary step before the compilation because, it deals with code manipulation and preparation tasks that are fundamental to creating modular, effective, efficient and a sustainable code.

Compiling:-

- Consider a program `first.cpp` which is stored in HDD.
- To compile this program, we need IDE that contains a compiler.
- The compiler will convert the high level code into machine level language code and a new executable file with name `first.exe` will be generated and get stored in the HDD.
- If the compiler finds the error in the code it throws the error to the programmer else the code is successfully compiled.
- The compiler translates the C++ program into machine language code (also referred as the object code).

* Sometimes assembler is used as an intermediate step.

Linking:-

The linker will combine all the external programs (such as libraries and other shared components) with our program to create a final executable file.

- it takes all the object files as input { creates a executable file.
- it resolves memory references.
- merges the object file

the console cursor is also called a carriage.

Basic terms.

printf \Rightarrow output funct.
cout \Rightarrow " ". Date: 20
MTWTFSS

/*, /*, // \Rightarrow comments.
are preprocessing directives.
<iostream> \Rightarrow input / output operations.

int main() \Rightarrow that an integer value is returned to the program.

return \Rightarrow to exit a program.
return 0: \Rightarrow the program terminates normally.

= (assignment operator).

std:: \Rightarrow specifies namespace, usually cout & cin belong from it.

>> \Rightarrow stream extraction operator.

<< \Rightarrow stream insertion operation.

\ \Rightarrow escape character that tells that a special character is to be outputted.

Escape Sequence:-

they are always started by a \ & used to format an output.

(i) \a :- produces alert bell.

(ii) \n :- new line (4 or 8 space & IDE)

(iii) \t :- tab. (gives space b/w two particular statements)

(iv) \\ :- it prints backslash and lets the compiler not confuse it with a program code.

(v) \" :- to print double quotes "

(vi) \b :- Backspaces

(vii) \r :- Carriage returns to the stat \Rightarrow current line

ASCII 0 - 48
 1 - 49
in 16 Bit (int \rightarrow 2 byte) in 32bit
 (int \rightarrow 4 byte)

2 - 30
3 - 51.

Date: 20
MTWTFSS

LECTURE NO.8

My notes:-

constants &
Data items that do not change value during the execution of a program.

Examples.

'A' '\$' '#' // character const.
"Hello" "a" // string const.
12 // integer const.
3.14 // floating point const.

(double commas are imp)

variable.

- location in memory (RAM) where any value can be stored.

- they must be declared with a name and a datatype. (before use)

example

int myvariable;
 ^
datatype name

- The value stored can be changed.

* Variables of same type can be defined;

(i) in same statement

e.g int length, int width;

(ii) in diff statement

e.g int length;
int width.

* Different variables must be defined in diff statements.

Keywords cannot be used as variable names.
Blank spaces cannot be used as variable names.
Special symbols, operator, punct. " " Date: 20
duplicate variable cannot be generated.

Understanding steps:- Digit cannot start a variable
32 letter is max for " ".

int n; → declaration.
n = 10; → Assignment.

if this is done altogether.

int a=5; → initializing (declaration +
Assignment).

PRIMITIVE DATATYPES:-

Integer data types:-

- (i) int : (2 byte) 0 - 4294967295
- (ii) long :- (4 bytes) 0 - 4294967295.
- (iii) short: (2 bytes) 0 - 65535

Decimal data types:-

float: - 4 bytes → 24-bit coefficient
double: - 8 bytes → 53 bit coefficient
→ 11 bit exponent.

Boolean :- /bool 1byte (true/false)
wchar - 1. 2-4 byte (1 wide character).
char 1byte (individual alphabets)

Understanding concepts:-

(i) int a=5;
now as int is equal to 4bytes i.e 32 bits;
a memory is allocated for the 5.

exceeding the limit
give us logical error.

1 byte = 8 bits
Date: 20
MTWTFSS
this is the most significant
bit as it will define if the no. is positive
or negative as (0 = + & 1 = -)

2/5
2/2-1
1-0

CAREFUL ANALYSIS:-

(i) float x = 10; (no syntax error as 10.0 is stored).

(ii) char ch = A;

(A has ASCII = 65 ⇒ 01000001)

also

char ch = 65;

hence both cases are same.

(iv) int i = 'A'

(no syntax error altho inverted commas represent it to be a symbol not an integer; as the memory is allocated in 0s & 1s).

(v) int i = 59.53

" now it is decimal not integer so the compiler will only convert 59 into object code and store it".

ARITHMETIC OPERATORS (FOR ARITHMETIC OPERATION = 5 Power has no ≈ in C++)

a. Addition (+)

b. Subtraction (-)

c. Multiplication (★)

d. Division

↙ ↘

Quotient

Reminder

(/)

(%) Page =
Modulus

using namespace std is a directive to skip writing std at every line of the text.

Date: 20
MTWTFSS

MTWTFSS

* BODMAS rule is followed thus parathesis are important to use.

RULES OF OPERATORS!

i) ()

parathesis are operated first.
in Nested paranthesis, the innermost is operated first.
if not nested it is operated left to right.

ii) * / %

they are operated 2nd and left to right if many.

iii) + or -

they are operated 3rd and left to right if many.

EQUALITY & RELATIONAL OPERATORS:-

>

$x > y$

x is greater than y

<

$x < y$

x is less than y

>=

$x >= y$

x is greater than or equal to y

<=

$x <= y$

x is less than or equal to y

==

$x == y$

x is equal to y

!=

$x != y$

x is not equal to y

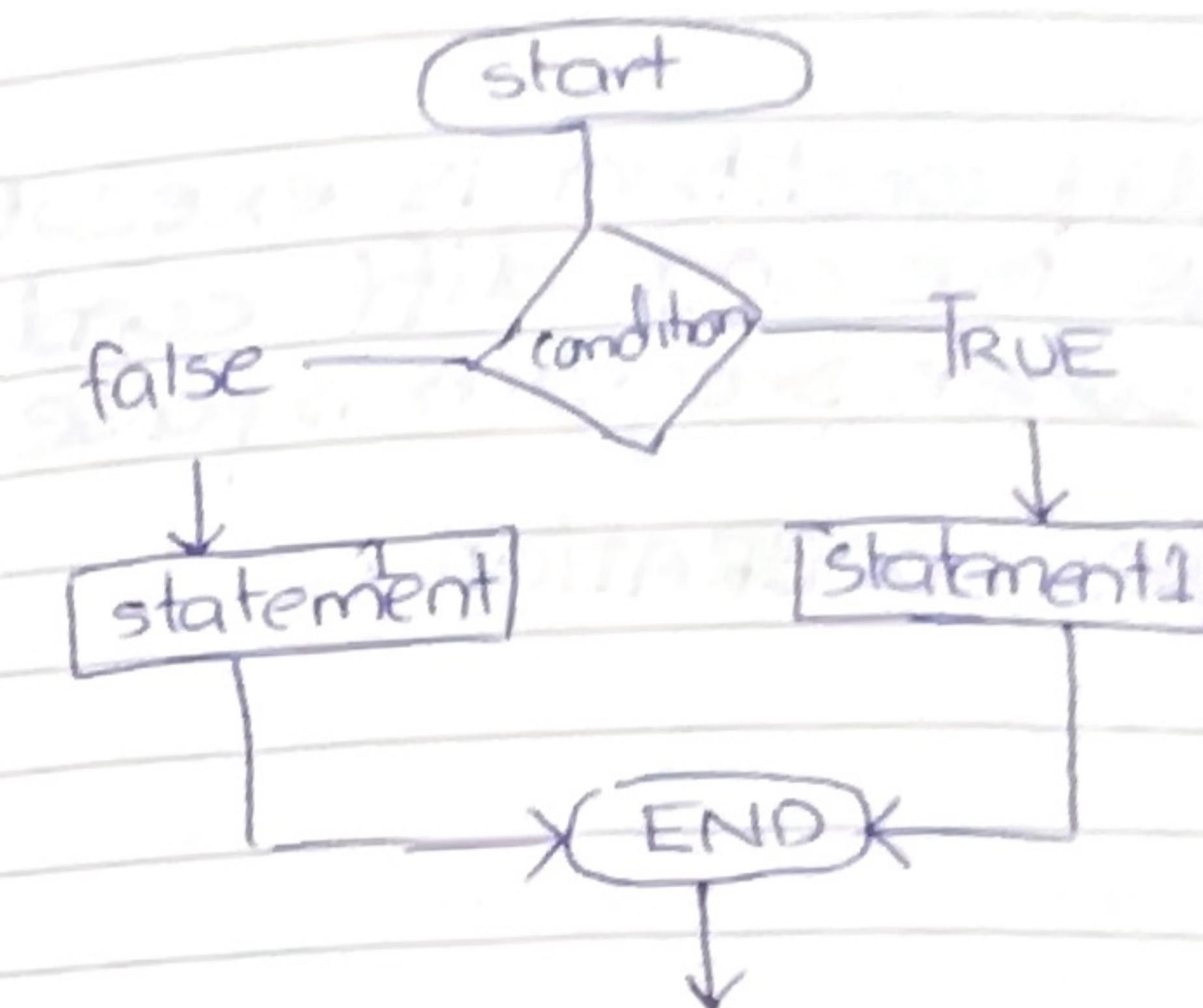
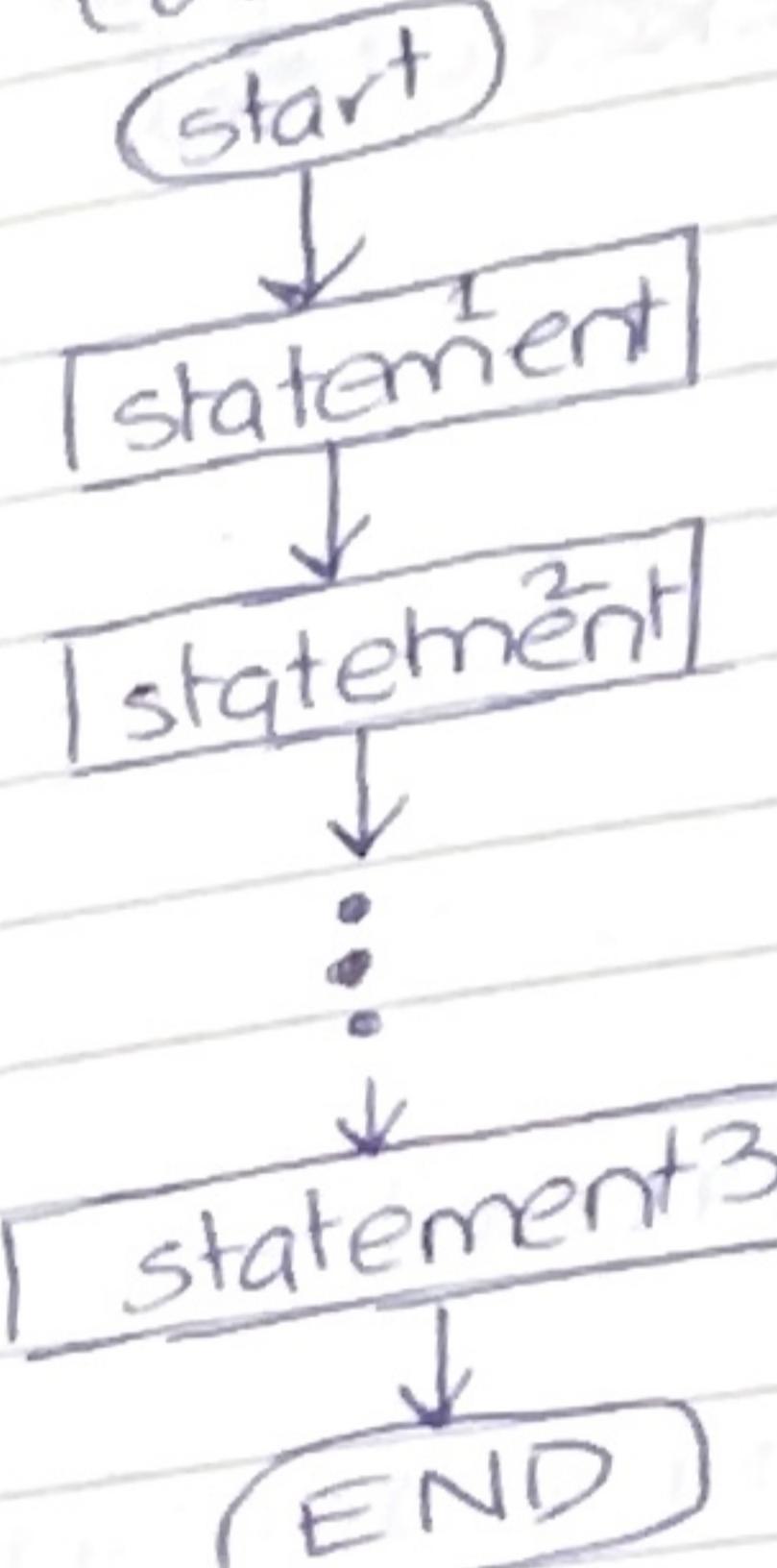
LECTURE : 9

(shapes a C++ program)

CONTROL STRUCTURE:-

structures that defines how a C++ code is compiled

a. SEQUENCE (default)



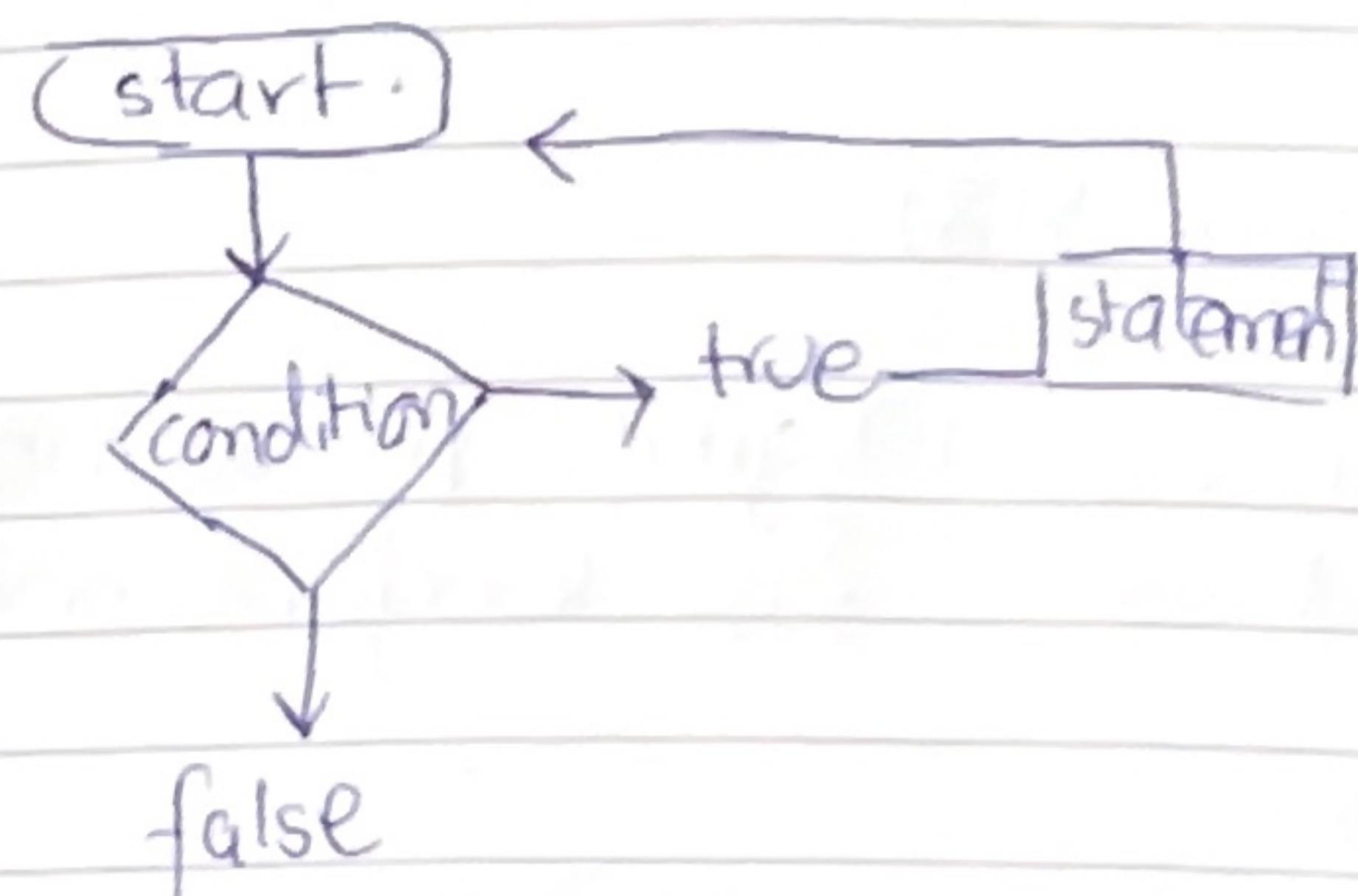
b. SELECTION

Example :-

if else, & if,
switch

c. REITERATIVE.

example:
while, do/while
and for.



(SINGLE ENTRY / EXIT)

- If selection structure is a one way selection structure that prints an output only if a certain condition is met.

- If / else selection structure is a two way selection structure i.e:-

diff condition is executed when program is true and diff cond. is executed when program is false.

CONDITIONAL OPERATOR:-

COMPOUND statement:- / block statement:-

A sequence of more than one statement bounded inside the curly brackets is called a compound statement.

```
{ if (age > 18)
```

```
    cout << "Eligible for voting" endl;
```

```
    cout << "No longer a minor" endl;
```

```
}
```

```
else (age)
```

```
{ cout << "Not eligible to vote" endl;
```

```
    cout << "still a minor" endl;
```

NESTED if else:-

Using if else conditions in the if else again (understand through this code):-

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

```
int main()
```

```
{ int var1, var2; → cin >> var1 >> var2;
```

```
if (var1 != var2)
```

if { cout << "var1 is not equal to var2" endl;

```
if { (var1 > var2)
```

cout << "var1 is greater than var2";

else { cout << "var2 is greater than var1".

```
}
```

```
else
```

```
{ cout << "var1 is equal to variable2".
```

```
}
```

```
return 0;
```

```
}
```

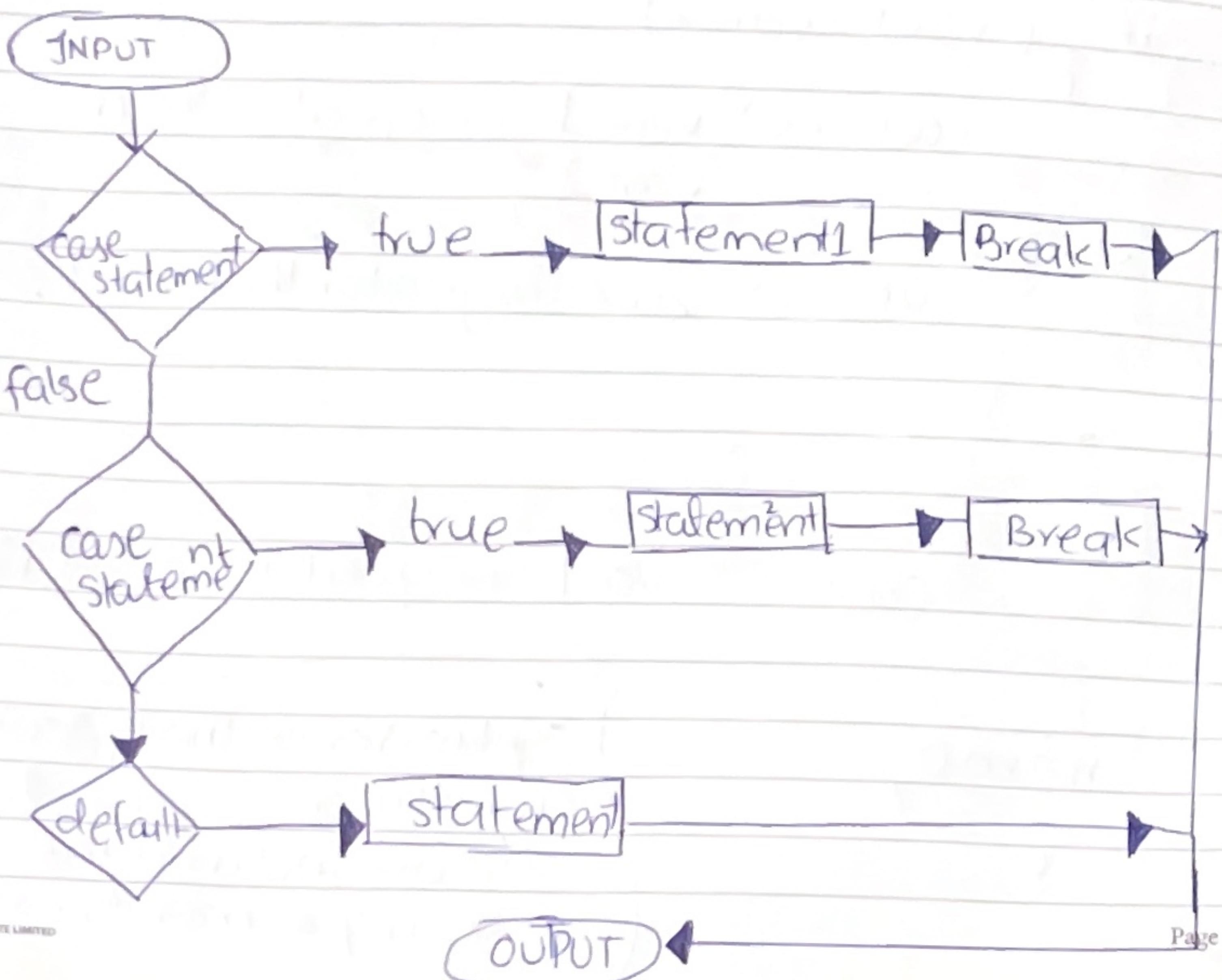
) Syntax error is the type of error that occurs when the code does not meet the coding requirements.

TYPES OF ERRORS :-

- (i) LOGIC ERRORS . (Runtime error) which affects the execution time.
- (ii) NON FATAL ERRORS.
- (iii) FATAL ERROR.

SWITCH STATEMENT:-

Using if else, we have only Binary selection. Sometimes it is necessary to branch in more than 2 directions. We do this via multiple selection ie "switch statement".



- Break statement:-**
- i) is a break statement
 - ii) allows immediate exit from a program (while/dowhile or switch)
 - iii) last statement in switch is almost always break .
 - iv) Cause (Break) program to jump to the closing braces of the switch structure.
 - v) if there is no break the program will follow into the next case i.e wasting time.
 - vi) default case is not compulsory for switch.

case statement:-

- i) No limitation of the case statements and assigning value as they can be assigned any no. / alphabet / characters (assigned)
- ii) They can be in any order ascending or descending.

LECTURES II

are that causes a statement to repeat

efficiently use variables.

basis are the the syntax the expression acts as a decision maker and usually using logic operation

oop).

loop as the expression is executed.

Now the loop will not execute as the condition is tested first and we see that

```

5)
    \n";
    ;
  
```

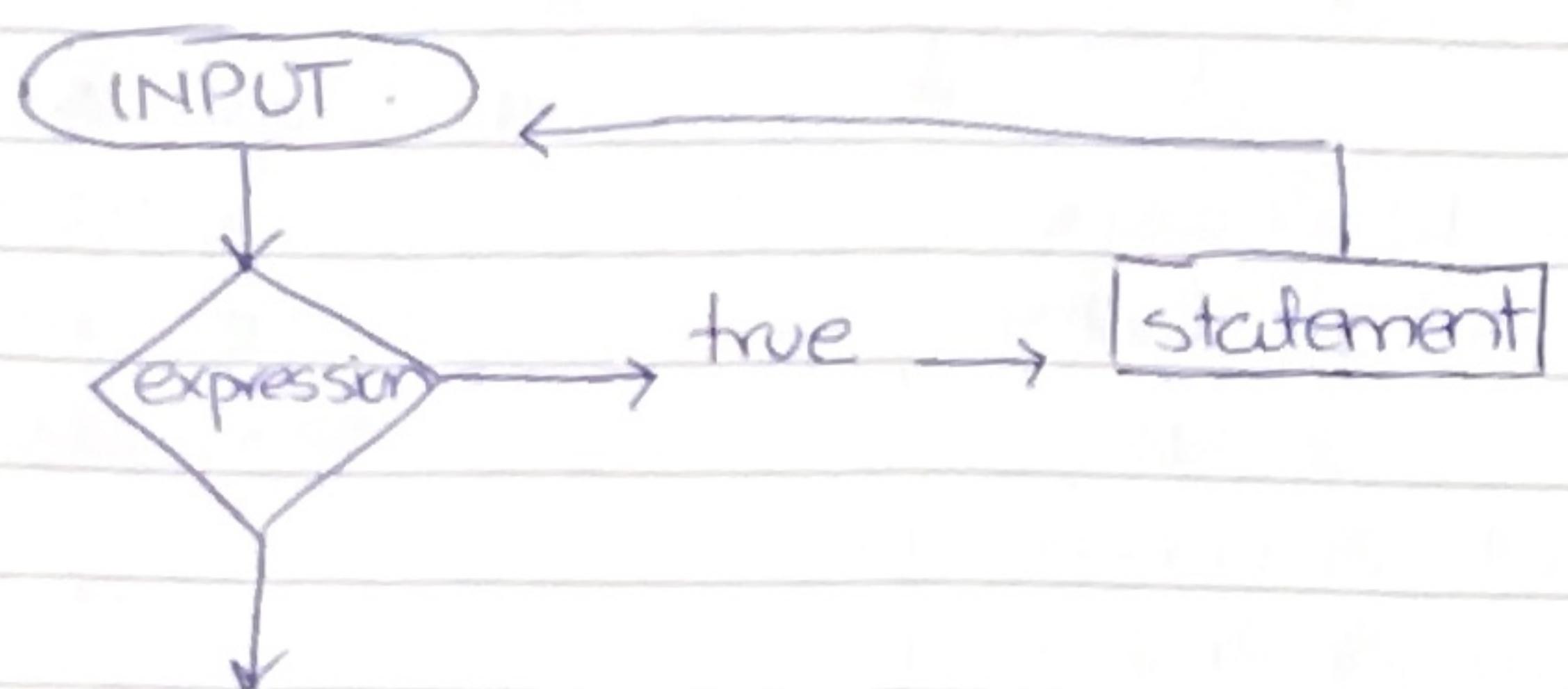
it is not satisfied.

Avoiding infinite loops:-

In order to avoid infinite loops.

- (i) The loop must contain code to make the "expression" false at some point (or the loop will not stop.) i.e infinite loop.

Flow chart of while repetitive structure:-



COUNTER CONTROL LOOP / REPETITION:-

In a while loop when we know how many times a certain code is to be executed it is called as a counter control repetition.

In a counter control repetition the number of repetitions are known. This structure requires

- (i) name of a control variable / loop counter.
- (ii) initial value of the control variable.
- (iii) loop continuation condition.
- (iv) incrementation / decrementation of control variables.

Example:-
A class of ten students took a quiz. The grades (0 - 100) for this quiz are available to you. Determine the class average of the quiz.

```

#include <iostream>
using namespace std;
int main ()
{
    int total ;
    int gradecounter;
    int grade;
    float average;

    total = 0 ;
    gradecounter = 1 ;
    while (gradecounter <= 10 )
    {
        cout << "Enter grade:" << endl;
        cin >> grade;
        total = total + grade;
        gradecounter = 1 + gradecounter;
    }
    avg = total / 10 .
    cout << "Class avg" << avg << endl;
    return 0;
}
  
```

SENTINEL CONTROL REPETITION:-

These types of repetitions are not controlled by a counter (expression / condition)

- Sentinel indicates "end of data entry".
- In such case the sentinel variable is tested in the expression / condition and the loop will end when the sentinel is encountered.

Example:

```
cin >> variable;
while (variable != sentinel)
{
    :
    cin >> variable
    :
}
```

Practical Example:-

```
#include <iostream>
using namespace std;
int main()
{
    int game = 1;
    int points;
    int total = 0;
    cout << "Enter the no. of points";
    cout << " Enter the points of the game" << game;
    cin >> point;
```

Date: 20
MTWTFSS

while (point != -1),

{ total += points; (or total = total + points)

game++;

cout << "Enter" << endl;

cin >> points;

}

cout << "The total points earned are" << total << endl;

return 0;

}

Do-while loop:- (Post test loop).

Do while loop unlike the while loop is a post test repetition structure because of the inclusion of do-statement which enforces C++ program to execute a certain code with any condition fulfillment.

Syntax:-

```
do
statement;
while (expression);
{
    statement;
}
```