

* Algorithm → In mathematics & computing an algorithm is a procedure for a complete task which is given an initial state.

→ Advantage →

- (1) It is independent of programme lang.
- (2) It is easy to debug.
- (3) Each step of algorithm can be easily coded into high level language.
- (4) Easy to understand and modify.

→ Requirement component of algorithm →

- (i) Definiteness
- (ii) Input
- (iii) Output
- (iv) Effectiveness

for example → Sum and average of three numbers. (Compulsory to write in capital bec. it is easy to identify steps)

Step 1 → INPUT A, B, C

Step 2 → SUM ← A + B + C

~~STEP 3 → AVG ← A + B + C~~

STEP 3 → AVG ← $\frac{SUM}{3}$

STEP 4 → PRINT "SUM is" SUM.

STEP 5 → PRINT "AVG is" AVG

STEP 6 → STOP

A Algorithm of maximum of two numbers.

STEP 1 → START

STEP 2 → INPUT A, B

STEP 3 → IF A > B

STEP 4 → PRINT "A".

STEP 5 → ELSE

STEP 6 → PRINT "B".

STEP 7 → STOP

* PSEUDO CODE →

- Pseudo Code is a programming analysis tool i.e. used for planning programme logic.
- Pseudo Code refer to true or false code. Each code are english like statement represent one logical operations of the programme being designed.
- Pseudo code is a ~~written~~ written statement of an algorithm using a restricted and well define vocabulary.

Advantage →

- (1) Converting a pseudo code to a programming language is much easier than converting a flow chart to a programming language.
- (2) Writing of pseudo code involves much less time than drawing an equivalent flow chart.
- (3) Pseudo code programmer can concentrate fully on the logic for the programme as rules are few and syntax is quit similar to programming language.
- (4) Modification of pseudo code is easier.

Ex - Sum and difference of two numbers →

INPUT Number -1

INPUT Number -2

SUM = Number -1 + Number -2

DIFFERENCE = Number -1 - Number -2

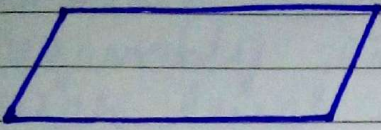
DISPLAY = SUM

DISPLAY DIFFERENCE

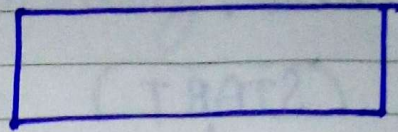
* Flow chart →

Representation of an algorithm and pseudo code in pictorial form.

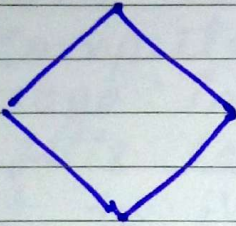
Symbols of flow chart →



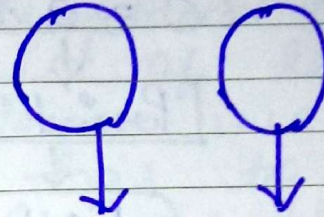
Input / Output



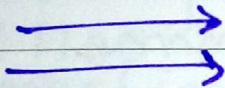
Processing



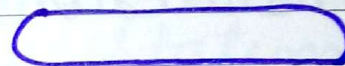
Decision making



Connectors



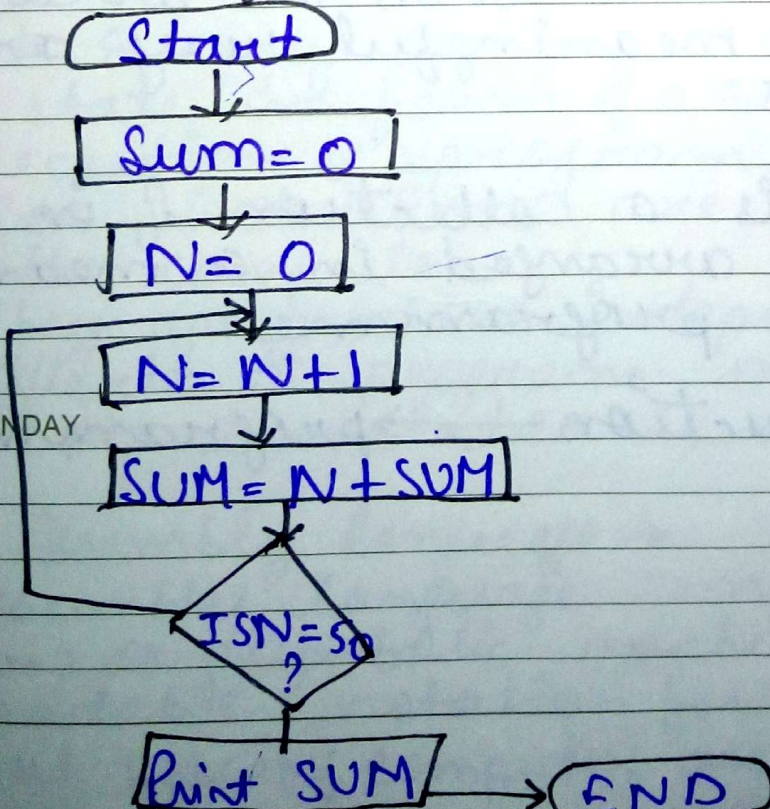
flow lines



Start / Stop

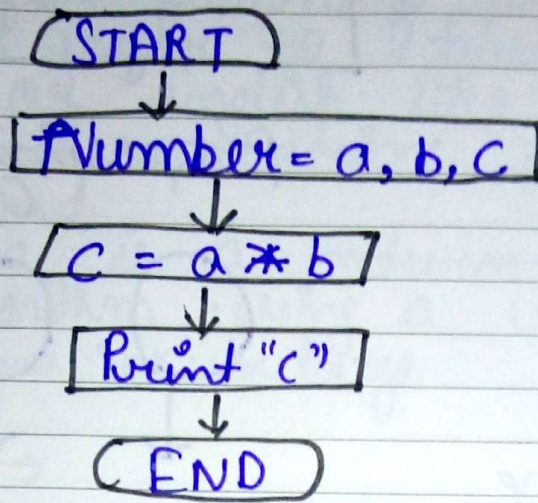
02

Ex → Draw a flow chart to find the sum of first fifty Natural Numbers.



03 SUNDAY

Q → Draw a flow chart for multiplication of two numbers?



★ Data instructions and programme

- A data → is any raw fact or raw material related with a physical or non physical entities like name of a person colour of chair salary of an employee etc.
- A instruction is made when we arrange the data in a meaningful way or order or steps
- A programme is a collection of no. of instructions arranged in a meaningful order called programme.

Data → instruction → programme

Classification of Computer Language

- ① Machine language [Binary $\rightarrow 0, 1$]
- ② Assembly language
- ③ High level language

* Machine language \rightarrow

Machine language is called the language of the computer that understand only this language in which consist of sequence of 0 & 1. Using this language, whatever we want to input to the computer has to be supplied in the form of the string as the 0 & 1. For example \rightarrow character 'a' may be represented as 01100101.

In machine language each statement given by user convert in binary format.

features of this language

- | | |
|----|----------|
| 07 | MAY 2015 |
| | THURSDAY |
- (1) It consists of 2 no. 0 & 1
 - (2) It is natural no. of computer.
 - (3) Internally 0 & 1 are converted into electric signals.
 - (4) Each & every command or operations must be state in form of 0 & 1.
 - (5) Execution of program is very fast as machine instructions are directly understood by the computer.
 - (6) They are machine independent.
 - (7) Difficult to program, modify & error prone.
 - (8) No translator is required.

* Assembly language \rightarrow

Assembly language commonly called assembly (asm) or symbolic machine code. is a human readable notation for the machine lang. that specify computer architecture uses.

Machine language a pattern of bits and coding. Machine operation is made readable by replacing the raw values with symbols that is called mnemonics code. For example -

```
MOV AL, 061H, 1011000001100001 D AX, ADD X,
MUL X, STA 1200H.
```

Features of this language →

- (1) Use of mnemonics code to write programs, which are small word specific to machine to denote a specific operation like add, sub, multiplication, which represent operation addition, subtraction, multiplication respectively.
- (2) For writing programs set of mnemonics code are made fixed for specific type of machine.
- (3) Assembler is an any system software which is specific for specific machine.
- (4) Easier to understand.
- (5) Easy to locate & correct errors.

★ High level language →

In assembly language and machine level language require deep knowledge of comp. hardware whereas in high level language we have to know only the instruction in this word & logic of program of computer are using.

High level language are simple language that used english & mathematical symbols like +, -, *, /, etc. for program instructions.

- features -
- (1) Program is written in simple english type language.
 - (2) Lets the program to concentrate on logic rather worry about hardware configuration.

- 3) Every instruction form a program is translated into Correspondance machine language program by which may be computer or interpreter.
- 4) Easy to use & modify
- 5) less error prone for each different HLL a separate compiler is required (translator).

Generation of programming language → Computer programming language can be categorised in fifth generation.

1) first generation → first generation is based upon machine language. Machine language is the only programming language that the computer can be understand directly without translation it is a language made up of entirely one and zero.

There is not however one universal machine lang. because the machine language must be returned accordance which be special characters of a given processor each type or family of processor or requires own machine language. For this reason machine language is said to be independent language in the computer 1st generation programmers had to use machine language because no other option was available.

(2) Second generation → The first step in making software development easier and more efficient was the creation in assembly language they are also classified as low level language because they knowledge of hardware is still required

They were developed in 1950 assembly language use mnemonic code expression and symbolic address in place of 1's and 0's to represent the expression codes. a mnemonic code is a alphabetical ~~expression~~ ^{abbreviation} used as memory aid it's means a programmer can use abbreviation. Initiative of having to remember lengthy, binary, initiation code.

for E.g. → It is much easier to remember L for load and A for add etc.

Assembly language uses symbolic address capabilities that specify the programming language processes because the programmer does not need to know or remember the exact storage locality or instructions or data. Symbolic address in terms of symbols chosen by the programmer rather than in terms of the absolute numerical location.

IIIrd generation →

language

The IIIrd generation, also known as high level language are very much like every day text and mathematical formulas in appearance they are design to run on a number of different computers with few or no changes. Most high level language are considered to be processor oriented or p language bec. the programmer instruction a list of steps processor that tell the computer not only what to do but how to do it, high level language statement's generation translated a comparatively greater number of assembly language instructions and even more machine language instructions.

developed software

A) IVth generation →
 IVth generation language

are also known as very high level language they are ~~also~~ non-procedural language so named because they allowed programmer and user to ~~specialised~~ specify what the computer is supposed to do without having to specify how the computer supported to do it. fourth generation language, ^{allow users or non comp. professional to} increase the speed of developing the properties in IVth generation lang. minimising user effort to obtain information from computer. In IVth generation language. five basic tools are also available :-

- ① Query language
- ② Report generation
- ③ Applications generator
- ④ Decision support system and financial planning languages.
- ⑤ Some micro computer application softwares.

* Feature of good programming →

- (1) Simplicity
- (2) Naturalness
- (3) Abstraction
- (4) Efficiency
- (5) Structured
- (6) Compactness
- (7) Locality
- (8) Readability
- (9) Portability
- (10) Error checking
- (11) Cost
- (12) Quick translation
- (13) Widely available

(1) Simplicity →

A good programming language must be simple and easy to learn and use.

26

MAY 2015
TUESDAY

It should provide a programmer with a clear simple and unified set of concepts which can be easy.

(2) Naturalness →

A good language should be natural for the application area for which it has been designed. i.e. it should provide appropriate operators, data structures, control structures and natural syntax to facilities the user to code their their their.

(3) Abstraction

Abstraction means the ability to define and then use complicated structures or operations in which that allow many of the detail to be ignored ^{ways}

27

MAY 2015

WEDNESDAY

TATA HITACHI

MAY 2015

S	M	T	W	T	F	S	S	M	T	W	T	F	S
						1	2	3	4	5	6	7	8
10	11	12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31						

(4) ~~Structured~~ → ^{Efficiency} written programmes written in a good programming language are efficiently translated into machine codes are efficient executed and acquire as little space in memory as possible.

(5) Structured —
Structured means that the lang. should have necessary features to allow its users to write their programmes based on the concept of Structured programming. This property of moreover it forces a programmer to look at a programme like any logical programme.

(6) Compactness —
In a good programming language programmers should be able to express intended operations concisely.

(7) Locality →
A good programming language should be such that while writing a programmer concentrate almost solely on the part of programme around the statement.

(8) Readability →
A good programming language will allow programmes to be written in same ways that resemble a quit english description of the underlying algorithms.

9) Portability →

A good programming language allow the writing of a wide variety of programmes. Thus, relatively the programmer of the need to become express in many languages.

10) Error checking -

A good programming lang. provide error checking facility acc. to programmer modular.

11) Cost →

A good programmer is that complete all programmes within time and cost.

12) Quick translation →

A good programming language allow quick translation facility to understand means of programming coding code.

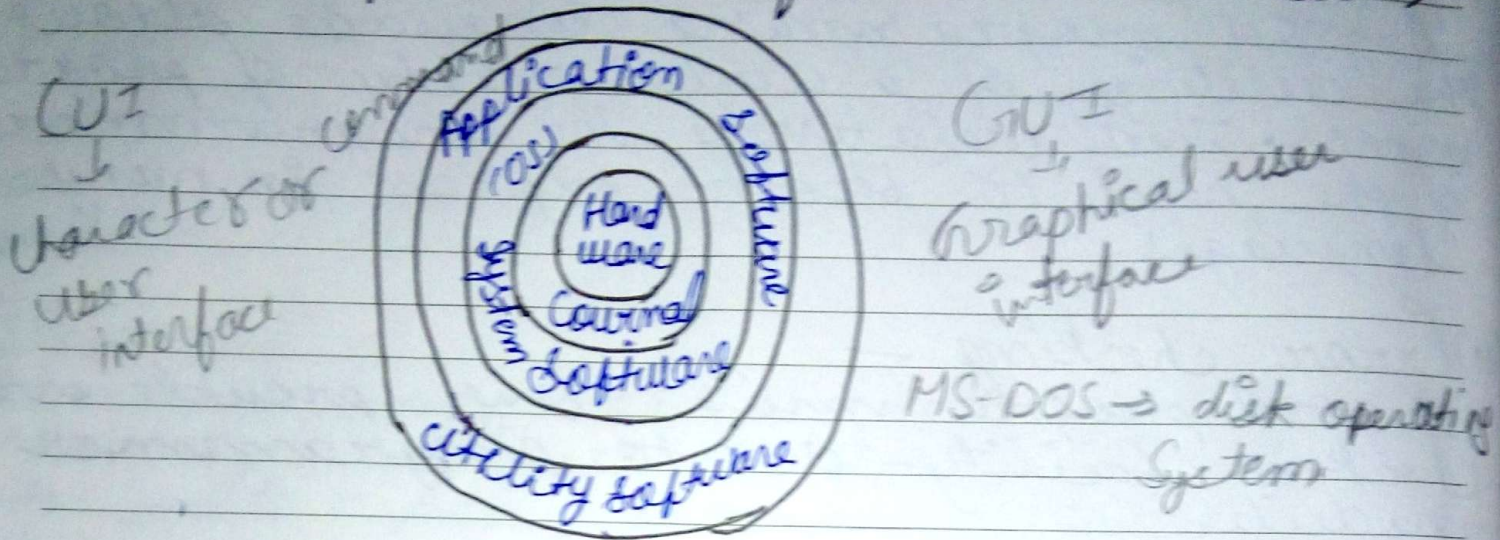
13) Widely available →

A good programming lang. provide all common code or syntax format or functions which are commonly and widely available.

* Computer Software →

A set of instructions (group of instructions) are called programme. & a group of programmes that cover more problems of user & solve diff. ways called software.

* Relationship between software and hardware →



* Software can be divided into 3 categories →

- (i) operating system or system software (windows)
- (ii) Application software

EXC) → windows 95, windows 3.1
 windows 98, windows xp,
 windows 2000, windows 2003
 windows nt, windows vista
 windows 7, windows 8
 windows 10

- System software is a software which is used to operate hardware according to user command.
- It is also known as a operating system or windows.

* Application software —

- Application software is used to after installation operating system or system software application software are available.

different package such as Ms office.

- MS-Office —
- (i) MS-Word
 - (ii) MS-Access
 - (iii) MS-Excel
 - (iv) power point
 - (v) outlook
 - (vi) Publisher

(2) Page Maker, (3) Photo Shop, (4) Corel draw } — DTP (desktop publications)
 Eg. Cover page of book, banner etc.

Wind RAR, Wind zip } — Virus cannot enter it at any cost.

- IsoBuster, Power Iso, Magic burn
- Narvow etc.

(3) Utility Software —>

Utility software is a system maintenance software in which we can manage data in harddisk remove virus from anti-virus, defragmentation etc. These are utility software.

arrangement of binary codes