

COMPUTER PROGRAMMING THROUGH C

UNIT :- 1

➤ ***DATA:- DATA IS A COLLECTION OF RAW FACT AND FIGURE, IS CALLED DATA.***

- Data is collected from different sources.
- It is collected from different purpose.
- Data may consist of numbers, characters, symbol or picture etc.

➤ ***INSTRUCTION:- A computer instruction is an order given to a computer processor by a computer program.***

➤ ***INFORMATION:- THE PROCESSED DATA IS CALLED INFORMATION.***

- Information is organized and processed from the data.
- It is more meaningful than data and used for decision making.

➤ ***PROGRAM:- A computer program is a sequence or set of instruction in a programming language for a computer to execute/run.***

➤ ***SOFTWARE:- Software, instruction that tell a computer what to do .***

- Software is a set of instructions, data or program used to operate computer and execute specific tasks.
- Software ex:- operating system , word processor , browser etc

There are two types of software.

1. SYSTEM SOFTWARE.

2. APPLICATION SOFTWARE.

What is Software in Hindi

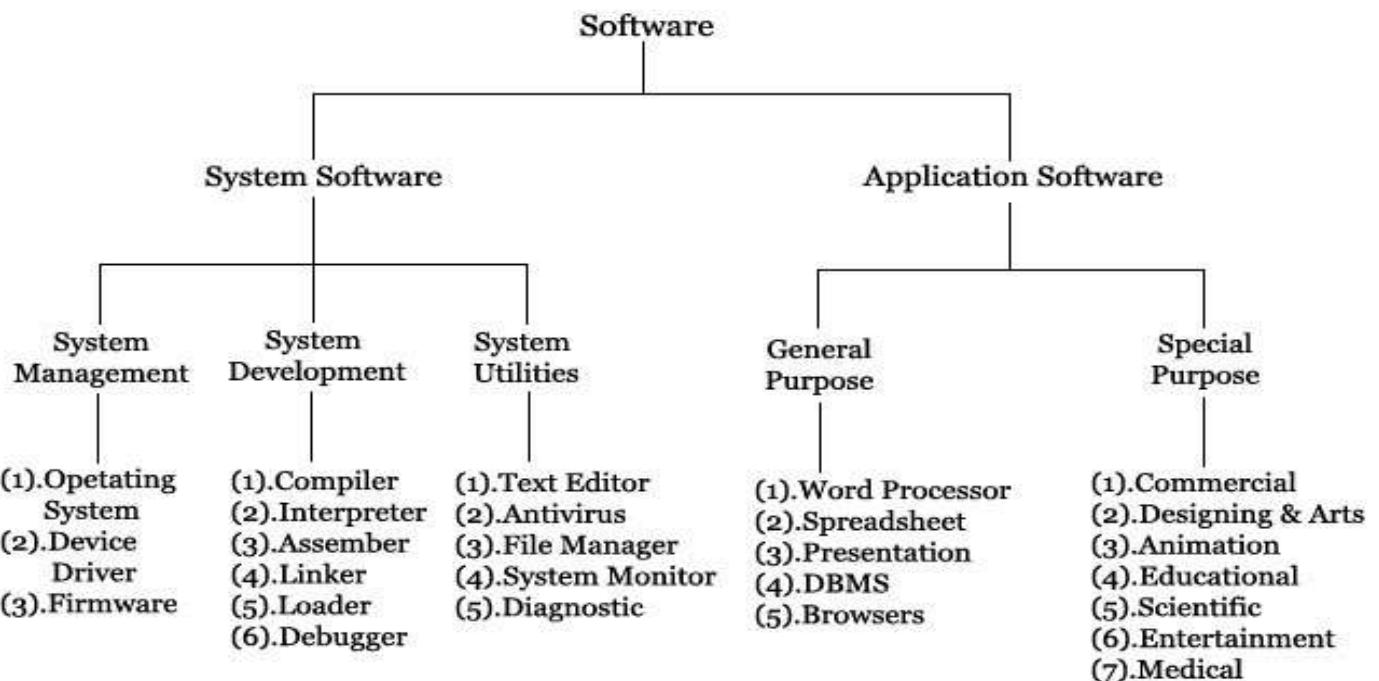


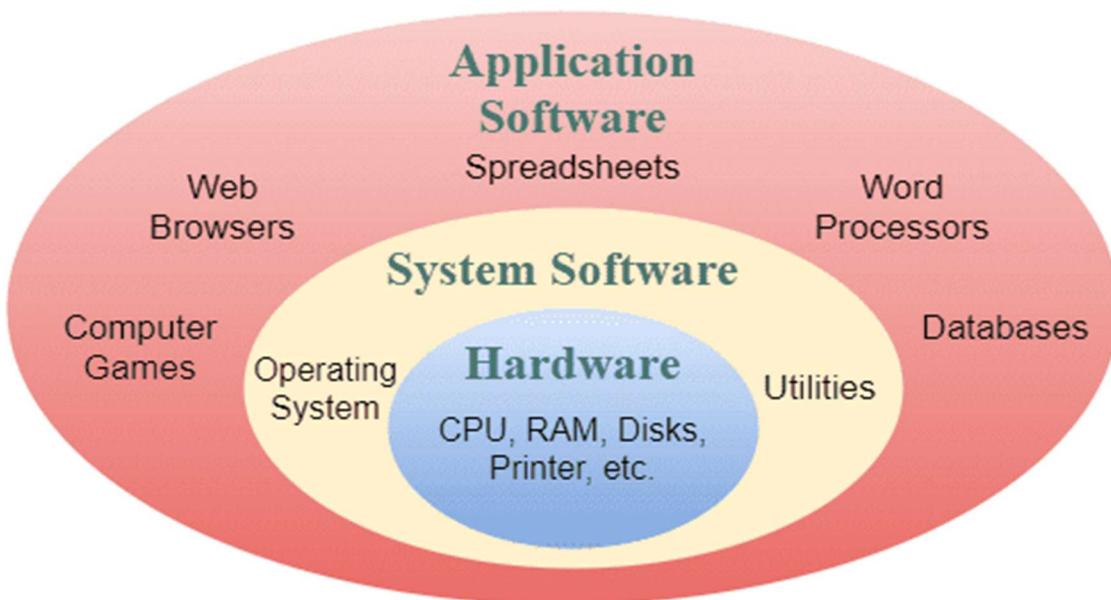
Fig: Classification of Software

- **SYSTEM SOFTWARE:-** *system software is a type of computer software that is designed to run a computer's hardware and application program.*
- System software is the type of software that is the interface b/w application software and system/ computer hardware.
- Low level language are used to write the system software.
- System software maintain the system resources and given the path for application software to run .
- An important thing is that without software , the system can not run .
- It starts when turn on the system ,stop when the system is turned off.
- System software programming is complex compared to application software .
- A computer can not read without system software.

- System software ex:- operating system , device driver ,firmware ,complier, assembler etc.

➤ ***APPLICATION SOFTWARE-:***

- Application software is that type of software that runs as per as user request.
- It run on platform which is provided by system software.
- High level language are used to write the application software.
- The main difference b/w system software and application software is that without system software the system can not run ,on other hand without application software the low level maintains system always runs.
- It starts a/c to the user's request .
- It programming is simpler compared to system software.
- Application software can not run without system software.
- Application software ex -: word processor , spreadsheet , browser.



-

LANGUAGE

- Language is a mode of communication that is used to share ideas, opinions and with each other.
- For ex :- if we want to teach someone ,we need a language that is understand by both communicators.



PROGRAMMING LANGUAGE

- A programming language is a computer language that is used by programmers (developers)to communicate with computers.
- It is a set of instructions written in any specific language (c , java , python)to perform specific task.
- A programming language is mainly used to develops desktop application ,websites, and mobile applications.

MACHINE LANGUAGE

- Machine language is a low- level programming language that includes and uses 0s and 1s .
- Computers are not capable of reading and understanding other language except machine language.

ADVANTAGE OF MACHINE LANGUAGE.

- ★ Program which are written in the machine language are quite efficient because it takes shorter time for execution as compare to those programs which are written in other languages.

- ★ No need of translation of the programs because those programs which are written in machine language are directly understood by the computers .

DISADVANTAGE OF MACHINE LANGUAGE

- ★ Machine language consist on 0 and 1 therefore all instructions written in the form of binary and numerical numbers .
 - ★ The numerical form of instruction is difficult to remembers and leads the errors.
 - ★ Errors are difficult to find. Each statement has to be carefully checked .
 - ★ When an error is found ,if and all of the statement below it have to be rewritten .
 - ★ The modification in machine language program is a difficult task because only modification in this language program result in a series of changes.
 - ★ If a program is to be modified by adding or deleting some instructions the addresses of all the subsequent instruction are to be changed.
 - ★ The part of previously written program in machine language can not be used in new program without changing the addresses of the instructions and data to confirm with the new program.
 - ★ The machine language is different for different system .this means that machine language is dependent.
 - ★ Any machine language program which is written for a particular computer can't be used on any other computer without drastic modification.

ASSEMBLY LANGUAGE

- ❖ Assembly language is a second generation language.
- ❖ An assembly language is a programming language that communicates with the hardware of a computer directly.
- ❖ This language assigns a **Mnemonic code** to each machine language instruction to make easier to remember or write.
- ❖ It allows better human readable method of writing program as compare to writing in binary bit patterns .
- ❖ Each processor family (and sometimes individual processor within a processor family) has its own assembly language.
- ❖ An assembly language provides a mnemonics instructions corresponding to each machine instruction.
- ❖ The basic unit of a assembly language is line of code .
- ❖ It allows the use of symbols and set of rule that can be used and compile to form a line of code .
- ❖ Each line of assembly language program consists of four columns called fields.
- ❖ The general format of a assembly language instructions is as follows.

[lable]	<Ocode>	<operands>	[;comments]
BEGIN	ADD	A , B	; ADD B TO A

MACHINE CODE	ASSEMBLY LANGUAGE	DESCRIPTION
001 1 000010	LOAD # 2	Load the value 2 into the accumulator
010 0 001101	Store 13	Store the value of the accumulator in memory location 13
001 1 000101	Load#5	Load the value 5 in the accumulator.
0100 001110	Store 14	Store the value of the accumulator in memory location 14.
001 0 001101	Load 13	Load the value of memory location 13 into the accumulator.
011 0 001110	Add 14	Add the value of the memory location 14 to the accumulator.
010 0 001111	Store 15	Store the value of the accumulator in any memory location 15
111 0 000000	halt	Store execution.

ADVANTAGE OF ASSEMBLY LANGUAGE

- ❖ The program written in assembly language are more easier to understand and use as compare to its machine language.
- ❖ assembly language is more user friendly language as compare to machine language.
- ❖ assembly language assembly language is easier to modify.
- ❖ Even in case of error , assembly language provides better facilities to locate and correct them as compare to machine language program .
- ❖ assembly language can run much faster and use less memory .

DISADVANTAGE OF ASSEMBLY LANGUAGE

- ❖ assembly language is not machine dependent .
- ❖ assembly language is harder to learn as compare to high level language .

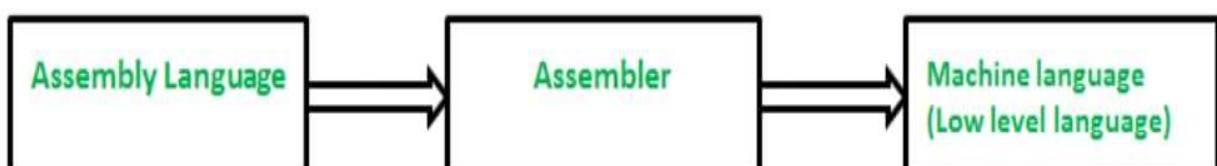
- ❖ assembly language, the development time can be 10-100 times as compare to a high level language .
- ❖ A program written in assembly language is less efficient than machine language because every assembly instructions has to be converted into machine language.
- ❖ assembly language needs to translator that is called **assembler**.

HIGH LEVEL LANGUAGE

- ❖ high level programming is written in natural language like English so that a common user could use the computer efficiently.
EX:- COBOL, FORTRAN , BASIC , C , JAVA , PYTHON, ETC.

ADVANTAGE OF HIGH LEVEL LANGUAGE

- ❖ It is a programmer friendly language.
- ❖ High level language is less memory efficient .
- ❖ It is easy to understand.
- ❖ It is portable.
- ❖ It can run on any platform.
- ❖ It is need complier or interpreter for translation.
- ❖ It is need widely for programming.



Computer Languages

Low Level Language (Machine Language)

Use 1's & 0's to
create instructions

Ex: Binary Language

Middle Level Language (Assembly Language)

Use mnemonics to
create instructions

Assembly Language

High Level Language

Similar to
human language

COBOL, FORTRAN, BASIC
C, C++, JAVA

C = A + B;
C C++ JAVA

High Level Language

ADD A , B Assembly Language

100100111 Machine Language

 Hardware

ALGORITHM

- The word is derived from the phonetic pronunciation of the last name of Abu, ja'far Mohammed ibn Musa al-Khowarizmi, who was an Arabic mathematician who invented a set of rules for performing the four basic arithmetic operation (addition, subtraction, multiplication, and division) on a decimal number.
- An algorithm is a step- by -step procedure to solve a given problem.
- An algorithm is a finite set of instruction, which is written for solve any problem.
- Algorithm is not the complete code or program, it is just like a English language.

EX:-

- 1) Take cap off tube of toothpaste.
- 2) Pick up toothbrush from holder
- 3) Put pea-sized amount of toothpaste on bristles of toothbrush.
- 4) Brush bottom teeth for two minutes.
- 5) Brush top teeth for 2 minutes.
- 6) Split out toothpaste.
- 7) Rinse toothbrush with water from tap.
- 8) Replace toothbrush in holder.
- 9) Replace cap on tube of toothpaste.



Ex :- **Algorithm for finding the average of three numbers.**

- Start
- Read 3 numbers a, b, c.
- Compute sum=a+ b+ c.
- Compute average=sum/3
- Print average value.
- Stop.

Ex :- **Algorithm for finding the area of rectangle.**

- Start
- Read two numbers a, b.
- Compute area of rectangle= a*b.
- Print area of rectangle.
- Stop.

ADVANTAGE OF ALGORITHM

- It is a step wise representation of a solution to a given problem, which it easy to understand.

- An algorithm uses a definite procedure.
- It is not dependent on any programming language, so it is easy to understand for anyone even without programming language.
- Every step in an algorithm has its own logical sequence so it is easy to debug.
- By using algorithm , the problem is broken down into smaller pieces or steps hence , it is easier for programmer to convert it into a actual program.

DISADVANTAGE OF ALGORITHM

- Written an algorithm takes a long time so it is time – consuming
- Understanding complex logic through algorithm can be vary difficult.
- Branching and looping statement are difficult to show in algorithm.

FLOW CHART

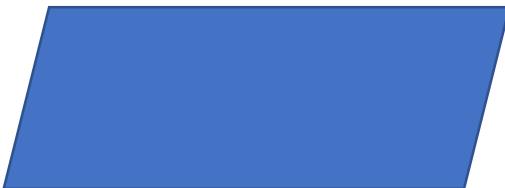
- A flow chart is a pictorial/graphical representation of an algorithm.
- The process of drawing a flow chart is known as “flowcharting”.

Basic symbol used in flow chart designs.

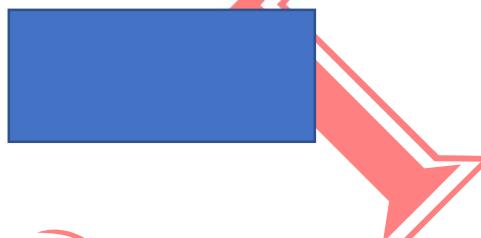
Terminal, process, input/output, decision, connector, and predefined process.

- **TERMINAL:-** The oval symbol indicates start, stop, and halt in a programs logic flow. A pause/halt is generally used in a program logic under some error conditions. terminal is the first and last symbols in the flow chart.

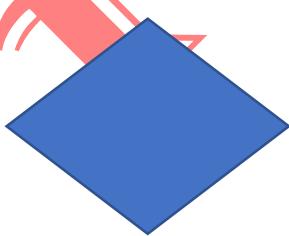
- **INPUT/OUTPUT**-: a parallelogram denotes any function of input/output type program instructions that take input from input devices and display output on output devices are indicates with parallelogram in a flow chart.



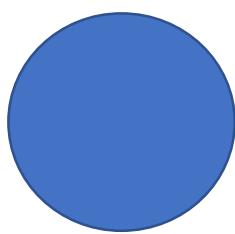
- **PROCESSING**-: A box represents arithmetic instruction. All Arithmetic processes such as adding, subtracting, multiplication and division are indicates by action or process symbol.



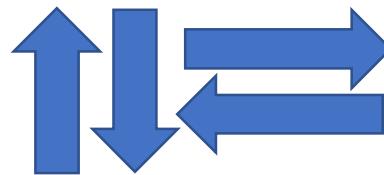
- **DECISION**-: Diamond symbol represent a decision point. Decision based operations such as yes/no question or true/false are indicates by diamond in flowchart.



- **CONNECTORS**-: Whenever flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusion. It is represented by a circle.



- **FLOW LINE**-: flow line indicates the exact sequence in which instructions are executed. Arrows represent the direction of flow of control and relationship among different symbols of flowchart.



Some rules while creating flowchart:-

Rule 1- Flow chart opening statement must be 'start' keyword.

Rule 2- Flow chart ending statement must be 'end' keyword.

Rule 3- All symbol in the flowchart must be connected with an arrow line.

	Terminator Indicates the beginning or end of a program flow in your diagram.		Subroutine Indicates a predefined (named) process, such as a subroutine or a module.		Connector Indicates an inspection point.		Collate Indicates a step that organizes data into a standard format.
	Process Indicates any processing function.		Preparation Indicates a modification to a process, such as setting a switch or initializing a routine.		Off-page connector Use this shape to create a cross-reference and hyperlink from a process on one page to a process on another page.		Sort Indicates a step that organizes items list sequentially.
	Decision Indicates a decision point between two or more paths in a flowchart.		Display Indicates data that is displayed for people to read, such as data on a monitor or projector screen.		Off-page connector		Merge Indicates a step that combines multiple sets into one.
	Delay Indicates a delay in the process.		Manual input Indicates any operation that is performed manually (by a person).		Off-page connector		Database Indicates a list of information with a standard structure that allows for searching and sorting.
	Data Can represents any type of data in a flowchart.		Manual loop Indicates a sequence of commands that will continue to repeat until stopped manually.		Off-page connector		Or Logical OR
	Document Indicates data that can be read by people, such as printed output.		Loop limit Indicates the start of a loop. Flip the shape vertically to indicate the end of a loop.		Summing junction Logical AND		Internal storage Indicates an internal storage device.
	Multiple documents Indicates multiple documents.						

ADVANTAGE OF FLOWCHART

- Flowchart are a better way of communicating the logic of the system.
- Flowchart act as a guide for blue print during program designed.
- Flowchart help in debugging processes.
- With the help of flowchart programs can be easily analysed.
- It provide better documentation.
- Flowchart serve as a good proper documentation.
- Easy to trace error in the software. Easy to understand.
- The flowchart reused for inconvenience in the future.
- It help to provide correct logic.

DISADVANTAGE OF FLOWCHART

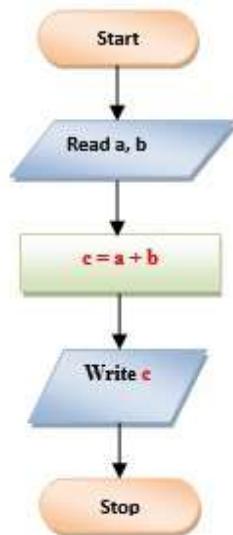
- It is difficult to draw flowchart for large and complex program.
- There is no standard to determine the amount of detail.
- Difficult to reproduce the flowchart.
- Making a flowchart is costly.
- Some developers think that it is waste of time.
- It makes software process low.
- If changes are done in software, then the flow chart must be redrawn.

To find sum of two numbers

Algorithm

1. Start
2. Read a, b
3. $c = a + b$
4. Print or display c
5. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int a, b, c;

    printf("Enter value of a: ");
    scanf("%d", &a);

    printf("Enter value of b: ");
    scanf("%d", &b);
    c = a+b;

    printf("Sum of given two numbers is: %d", c);

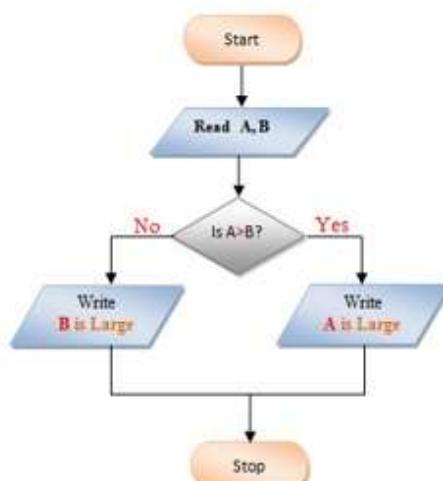
    return 0;
}
```

Greatest of two numbers

Algorithm

1. Start
2. Read A,B
3. If $A > B$ then
Print A is large
else
Print B is large
4. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int A, B;

    printf("Enter values of A, B: ");
    scanf("%d %d", &A, &B);

    if (A>B)
        printf("A is Larger");
    else
        printf("B is Larger");

    return 0;
}
```