Module 1 - chapter - 1

C Programming for Problem Solving

Unit-1 - Introduction to Computer Hardware & Software

→ Computer Types
→ CPU, Primary & Secondary memory
→ Input & Output devices
→ Software - types.

Def" of a Computer :-

A Computer is an electronic device that takes data & instructions as an input from the user, processes data, & provides useful information known as output.

* The electronic device is known as hardware.
* The set of instructions is known as software.

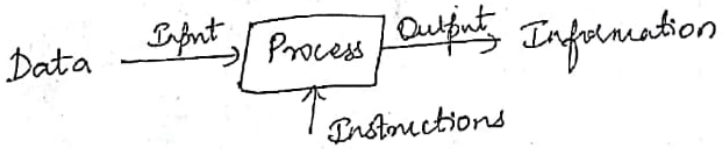
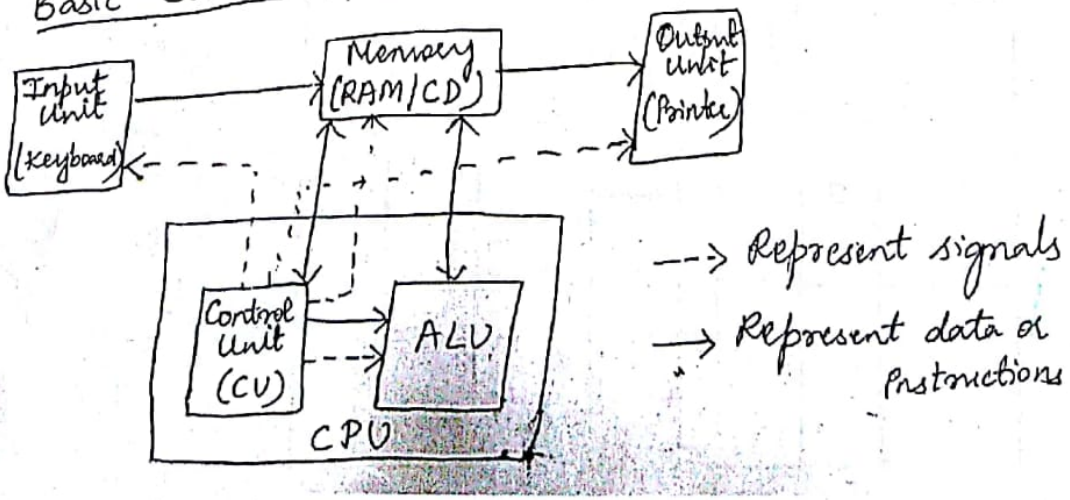


Fig. Input - process - output Concept

Basic Structure / functional units of a computer



Functional block diagram of a Computer

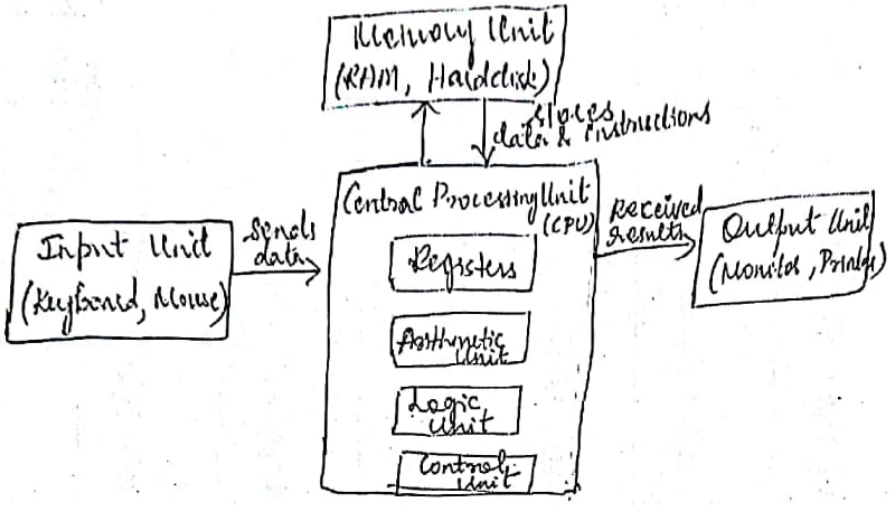


Fig. Essential Computer hardware.

* Input Unit
→ Accepts data on which operations are to be performed.
→ Comprises of various i/p devices such as keyboard, mouse, trackball etc.

* Output Unit
→ It provides output obtained after performing operations on the input data.
→ comprises of various o/p devices such as monitor, printer, plotter etc.

* Central processing Unit (CPU)
→ It is the brain of the computer that actually runs program instructions & processes the i/p data to generate desired output.

→ Comprises of following subcomponents,

- Arithmetic Unit - performs arithmetic operations on the input data.
- Logic Unit - Performs logical operations on the input data.
- Control Unit - Controls program execution by fetching instructions from memory, decoding & executing the instructions & delivering output to the users.
- Registers - Storage elements present inside the processor for storing intermediate results.

* Memory Unit

→ It stores the input data & output generated by a program.

→ Typical storage devices included in a memory unit are RAM, hard disk, floppy disk etc.

Input devices

→ An interface between the computer & the user.
→ It is used to take input from the user, translate it into the machine-readable form & send it to ~~processing~~ CPU for execution.

Some of the typical i/p devices are,

① Keyboard
② Mouse
③ Trackball

① Keyboard

→ Includes *Alphanumeric keys - number keys & alphabet keys
* Function keys - used to perform a specific task such as searching a file or refreshing a Webpage
* Modifier keys - such as Shift & Control keys modify the casing style of a character/symbol
* Cursor movement keys - up, down, left & right keys to modify the direction of the cursor on screen
* Other keys on keyboard includes spacebar, escape key, special keys such as Home, Insert, Delete & End.

② Mouse

→ The mouse allows the user to select elements on the screen, such as tools, icons & buttons, by pointing & clicking them.
→ Used to draw & paint on computer screen.

→ Also known as 'Pointing device' because it helps change the position of the pointer or cursor on the screen.

→ Mouse consists of,
* Left button - used to select an element on screen.
* Right button - displays special options such as open, explore & shortcut menus.
* wheel at top - used to scroll down in a document.
* ball at bottom - the cursor on the screen moves in the direction in which the ball rotates.

③ Trackball

→ Is a pointing device that consists of a socket containing a ball, rolled manually to move cursor on screen.

→ Socket contains sensors, which detect the movement of the ball.

→ Two types,
* Small trackball - used in portable computers
* Large trackball - for CAD, desktop systems.

→ Used as game controller in ~~some~~ computer games.

④ Bar Code Reader

⑤ Joystick

⑥ Trackpoint.

Output devices

→ Def" - Output devices pass on the processed data i.e, information through different mediums to the end users.

Some of the output devices are,

① Monitor
② Printers
③ Plotter
④ Speakers
⑤ Data projectors

① Monitor

→ most commonly used o/p device that produces visual displays generated by the computer.
→ used for visual presentation of textual & graphical information.
→ classified as,
* cathode ray tube (CRT) monitors are large, occupy more space.
* liquid crystal display (LCD) monitors are thin, light weighted & occupy lesser space.

② Printer

→ An output device that is used to produce a hard copy of the electronic text displayed on the screen into a paper sheets that can be used by an end user.
→ Printer driver software is used to convert a document to a form understandable by the computer.

→ Performance of printer is measured in terms of dots per inch (DPI) & pages per minute (PPM) produced by the printer.

→ Greater DPI - better quality.
Higher PPM - higher efficiency.

→ Classification of printers based on technology they use to print text & images,

Impact printer

ⓐ Dot matrix printers

Use perforated sheet to print the text. The process to print a text involves striking a pin against a ribbon to produce its impression on the paper.

ⓑ Inkjet printers

→ Slower than dot matrix & are used to generate high quality photographic prints.

→ Ink cartridges are attached to printer head that moves horizontally, from left to right.

Non-Impact printers

ⓒ Laser printers

→ may or may not be connected to a computer, to generate an output.

→ Consists of a microprocessor, ROM & RAM, which can be used to store the textual information.

→ The printer uses a cylindrical drum, a toner & the laser beam.

③ Plotter

→ An output device connected to a computer to print large documents such as engineering or constructional drawings.

→ classified based on their performance,

ⓐ Drum plotter - used to draw perfect circles & graphic images.

- drawing arm is used to draw the image.
- it rolls paper back & forth through roller & drawing arm moves across paper.

ⓑ Flat-bed plotter - has a flat drawing surface & two drawing arms that moves across paper to draw an image.

→ low speed & large in size.

ⓒ Inkjet plotter - Spray nozzles are used to generate images by spraying droplets of ink onto the paper.

→ High maintenance cost.

ⓓ Electrostatic plotter - produces quality print with highest speed.

→ It uses charged electric wires & special dielectric paper for drawing.

④ Speaker

→ Speaker is an electromechanical transducer that converts an electrical signal into sound.
→ provide audio output such as warning sounds & Internet audios.
→ Built-in speakers are also present that warn end users with error audio messages & alerts.
→ Subwoofer unit can be used to enhance bass output.

⑤ Data projectors

→ A projector is a device that is used for projecting an image onto the big white screen.
→ Images projected are larger in size compared to original images.
→ It consists of an optic system, light source & displays, which contain the original images.

Three types of projectors are,

ⓐ Ultralight/portable projectors - small in size.
ⓑ Conference room projectors - for projecting output at buisness meetings & huge gatherings.
ⓒ Fixed installation projectors - fixed permanently at the place of presentation, capable of projecting images in bright light also.

→ They are expensive.

Central Processing Unit [CPU]

* It is the brain of the computer that is responsible for controlling & executing program instructions.

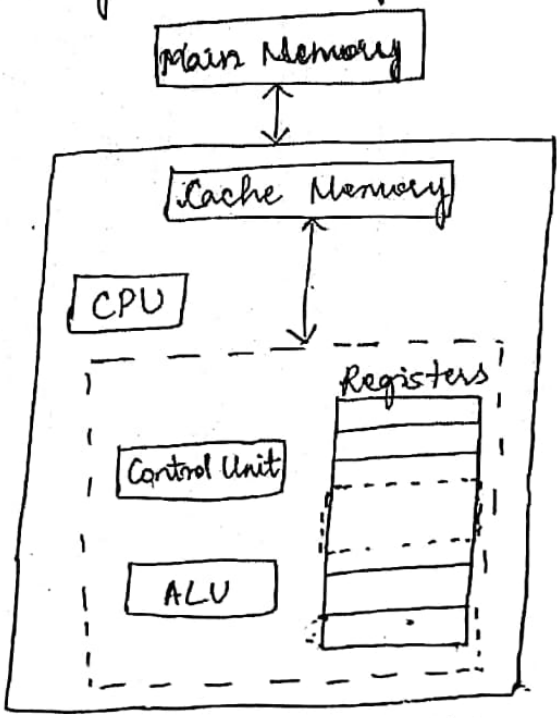


Fig. Illustration of CPU.

* CPU consists of following subsystems,

ⓐ Arithmetic logic unit (AU).
ⓑ Logic Unit (LU)
ⓒ Control unit (CU)

ⓐ Arithmetic Unit

→ performs arithmetic operations on the data, like addition, subtraction, multiplication or division.

(b) Logic Unit

→ performs logical operations on the data.
→ It performs 16 different types of logical operations like greater than (>), less than (<), equal to (=), ≠, shift left, shift right etc.

(c) Control Unit

→ CU is an important component of CPU that controls the flow of data & information.
→ maintains sequence of operations performed by the CPU.
→ It fetches an instruction from storage area, decodes the instruction & transmits corresponding signals to AU/LU & storage registers.
→ CU guides AU & LU about operations to be performed & suggests I/O devices to which data is to be communicated.

Cache Memory

→ is a small, fast & expensive memory that stores the copies of data that needs to be accessed frequently from the main memory.
→ Processor, before reading data from or writing data to main memory, checks for the same data in the cache memory.
→ If it finds, the data processor reads/wrote data to the cache itself because access time is much faster than main memory.

Registers

→ Special purpose, temporary storage units.
→ High-speed memory locations used for holding instructions, data & intermediate results.
→ Diff types of registers to hold diff types of "info",

(i) Program Counter (PC) – keeps track of the next instruction to be executed.

(ii) Instruction Register (IR) – Holds instruction to be decoded by the control unit (CU).

(iii) Memory Address Register (MAR) – Holds the address of the next location in the memory to be accessed.

(iv) Memory Buffer Register (MBR) – Stores data received from or sent to CPU.

(v) Memory Data Register (MDR) – Stores operands & data.

(vi) Accumulator (ACC) – Stores the results produced by arithmetic & logic units.

Primary and Secondary Memory

Memory – Definition :-

Memory is a place / locations where the data and/or the instructions are stored to achieve a specific task.

→ Memory can be broadly classified as,

Memory
- → Primary Memory (Main) — Eg, RAM, ROM
- → Secondary Memory (Auxiliary) — Eg:- Floppy disk, Hard disk, CD, Magnetic tapes.

Primary Memory

Primary Memory
- → Volatile — Eg:-RAM
- → Non-volatile — Eg:- ROM, Flash memory etc.

Def^ of Primary Memory

The memory that is accessible directly by the CPU of a computer.

* Primary memory allows the CPU to store & retrieve data quickly.

Advantages - (i) Data/instructions accessed at very high speed

(ii) Before execution of program, all instructions/data used by CPU have to be loaded into main memory.

Disadvantages

(i) Primary memory is volatile.

(ii) Very expensive.

(iii) Huge amount of data cannot be stored.

Types of Primary memory:-

(i) Volatile Memory

* The memory that loses its contents when the computer is turned off is called volatile memory.
* Also called read-write memories, since CPU can read the data & write the data into the memory.

Eg:- DRAM (Dynamic), SRAM (Static).

RAM (Random Access Memory)

* Semiconductor memory.
* Data & programs that are entered from keyboard are stored temporarily in RAM during execution.

(ii) Non-Volatile Memory

* The memory that retains its contents, even after the computer is turned off is called non-volatile memory.
* Hold data permanently.
* Only read operations are allowed.
* Contents of these memories cannot be changed normally.

Eg:- ROM, PROM (programmable)
(Read only Memory)

Secondary Memory

→ The memory that can store all the data & instructions even when the computer is turned off.
→ Secondary storage is used to store data & instructions permanently.
→ Data & instructions are loaded from secondary memory to main memory for CPU to process data.

Adv:-

(i) They are Portable.
(ii) Very cheap.
(iii) Huge amount of data can be stored.
(iv) Data stored in the secondary memory are not lost even when computer is turned off.

Disadv:-

* Accessing of data is very slow.

Difference between RAM & ROM :-

RAM	ROM
① Data can be read & written into RAM.	① Data can be read (read only) but data cannot be written into ROM.
② Volatile - contents of RAM are lost when computer is off.	② Non-Volatile - contents of ROM are not lost when computer is off.
③ Temporary storage.	③ Permanent storage.
④ Programs are brought into RAM just before execution	④ Programs to be executed are already available in ROM (during manufacturing time).

Difference between Primary & Secondary Memory :-

Primary Memory	Secondary Memory
① It is an internal memory also called Main memory.	① External memory of the computer.
② Accessing data is fast.	② Accessing data is slow.
③ Expensive.	③ Less expensive.
④ Temporary Storage.	④ Permanent storage.
⑤ Huge amount of data cannot be stored.	⑤ Huge amount of data can be stored.
⑥ Semiconductor memory	⑥ Magnetic memory.

Software Types:-

* Software is a set of programs / logical instructions written in computer programming language that tells the computer how to accomplish a particular task.
* Software enables proper functioning of hardware components & perform their respective operations effectively.

Two types ;-

(a) System software

The programs, which control & co-ordinate the hardware components & manage their performance.

Eg:- Operating Systems, compilers, assemblers, interpreters, device drivers etc.

(b) Application software

The programs, which are designed to meet the information processing needs of the end users.

Eg:- Word processors, database programs, banking software, MS-word, Excel etc.

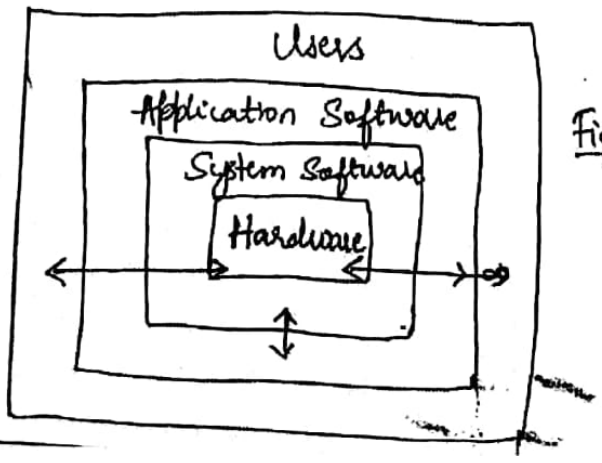


Fig. Layers of software & their interactions.

* System s/w also allow the application s/w to interact with the underlying hardware, thus acting as an interface between them.

Computer Types :-

* Methods or techniques used by computers to process & handle data may be different. Hence, computers are classified into,

1. Based on Operating principles
   - Analog
   - Digital
   - Hybrid
2. Based on Applications
   - General-purpose
   - Special-purpose
3. Based on Size & Capability
   - Microcomputers
   - Mini computers
   - Mainframe computers
   - Super computers

(1) Based on Operating principles

On basis of operations performed & methods used to store & process data & information, it is classified into,

* Analog Computers

→ Represents data in the form of continuous electrical signals having a specific magnitude.

→ Very fast in operation, allows several operations to be carried out out the same time.

→ Results produced are not very accurate.

→ Powerful tools to solve differential equations.

* Digital Computers

→ Also called digital information processing system, that stores & processes data in the digital form, i.e, 0's & 1's.

→ Digital computers are also capable of processing analog data.

→ Faster, reliable & produce more accurate results

Eg:- Peesonal computer.

* Hybrid Computers

Hardware components of hybrid computers are a

→ Combination of analog & digital computer components

→ Accepts analog signals & convert them into digit form.

Very fast, efficient & reliable.

Cost-effective in performing complex simulations

computer in hospital to measure heartbeat of a patient, Robotics. etc.

② Based on Applications

On the basis of different applications or purposes, computers can be classified into,

* General-purpose computers

→ designed to work in all environments.
→ Versatile & can store a number of programs meant for performing distinct tasks.
→ Not efficient & consume more time to generate result.

Eg:- Laptop, Smartphones etc.

* Special-purpose computers

→ designed to perform only a specific task.
→ Not versatile, speed & memory depend on the task that is to be performed.
→ Efficient & consume less amount of time to generate result.

Eg:- ATM machine, systems that control military planes, defense-oriented applications etc.

3) Based on Size & Capability

* Micro computers

→ is a small, cheap digital computer, i.e, designed to used by individuals.

→ It consists of microprocessor, a storage unit & an I/O channel.

Eg:- PC's, workstations, notebook computers.

Depending on size, microcomputer is further classified as,

(i) Desktop Computer -

→ Also known as PC (Personal Computer).

→ designed to be used by an individual at a single location.

→ Components of a desktop computer are keyboard, mouse, monitor, hard disk storage, peripheral devic & a system unit.

→ Very cheap.

Adv - Powerful, Easy maintenance, Cheap, General-pr

Disadv - ~~Space~~ Occupy more space, Portability.

(ii) Laptop Computer - [Notebook / Notepad / Mobile comput

→ Portable computer that can be taken from o place to another at any time very easily.

→ Small in size.

→ Has rechargeable battery that removes the of continuous external power supply.

→ More expensive.

Adv - Portable, occupy less space, powerfull.

(iii) Hand-held computer (PDA's - Personal Digital Assistant)

→ It is a very small-size computer that can be kept in pocket.

→ Has a very small display screen, i/p device to this is a pen / electronic stylus.

→ Less storage capacity.

→ Less powerful compared to desktop & laptops.

* Mini-Computers

→ Are computers whose capabilities are between mainframes & microcomputers, hence called 'midrange computers'. (less powerfull than mainframe but more powerful than micro).

→ Caters the needs of multiple users at a single instant of time ranging between 4 to 200 users

→ Used as a network server, web servers (that can handle thousands of transactions in a day).

→ Less expensive than mainframe.

Eg:- PDP 11, IBM (8000 series), VAX 7500 etc.

* Mainframe Computers

→ Very large computer employed by large business organisations for handling financial transaction processing, Enterprise Resource Planning (ERP), indu censes etc.

→ handles millions of records in a day.
→ More expensive.
→ Implementation requires large space with monitored humidity & temperature levels.
Eg:- IBM 3000, VAX 8000, CDC 6600.

* Super Computers

→ Fastest type of computer that can perform complex operations at a very high speed.
→ More expensive.
Eg:- CRAY 3, Cyber 205, NEC SX-3 & PARAM from India

Applications areas of Super computers are:-

(i) Weather forecasting
(ii) Animated graphics
(iii) Fluid mechanics
(iv) Nuclear energy research
(v) Petroleum exploration.

* Introduction to C language
* Algorithm & flowchart
* Structure of C program
* C Tokens & data types

* Introduction to C language :-

C is a high level language. It was developed at the Bell Laboratories of USA in 1972. It is the outcome of efforts of Dennis Ritchie & Brian Kernighan.

Characteristic of C

* C language became popular because of the following reasons.

- C is a robust language, which consists of number of built-in functions & operators which can be used to write complex programs.
- C program is easy to learn.
- It helps in development of system software.
- No rigid format. Any number of statements can be typed in a single line.
- C is a structured programming language

- portability - C program can be run on different machines with little or no modifications.

Applications of C

C is used to develop the system as well as application softwares.

System softwares like,
- Operating systems
- Interpreters
- Compilers
- Loaders
- Linkers
- Editor

Application softwares like,
- Graphic package.
- CAD applications
- Scientific & Engineering applications.

* Algorithm & flowcharts :-

* Algorithm & flowcharts are the 2 important tools used in problem solving using digital computer.

* Algorithm & flowcharts & pseudo codes are some of the design techniques.

Algorithm

* Algorithm is a problem solving technique.
* An algorithm can be defined as a step-by-step procedure to solve a particular problem.
* It consists of english like statements. Each statement must be well defined to perform a specific operation. When these statements are executed for a given set of conditions, they will produce the required results.

Characteristics of algorithm

Each & every algorithm is characterised by the 5 important characteristics:

1. Input: It may accept zero or more inputs.
2. Output: It should produce at least one output (result).
3. Definiteness: Each instruction must be clear, well-defined & precise. There should not be ambiguity.
4. Finiteness: It should be a sequence of finite instructions.
5. Effectiveness: It means that the operations must be simple & carried out in a finite time.

Algorithm notations

While writing algorithm the following notations are considered.

1. Name of algorithm: It specifies the problem to be solved.
2. Step number: It is an identification tag of instruction. It is an positive integer.
3. Explanatory comment: It follows the step number & describes the operation. It should be written within a pair of square brackets.
4. Termination: It specifies the end of algorithm. It is generally "STOP" statement & last instruction of algorithm.

Example 1: Write an algorithm to compute area of a circle.

Algorithm: Area of circle
Step 1: Start
Step 2: Read radius
Step 3: [compute the area]
Area = 3.142 * radius * radius
Step 4: [print the area]
print "Area of circle =", Area
Step 5: [End of algorithm]
Stop

Example 2: Write an algorithm to performs the basic arithmetic operations such as addition, subtraction, multiplication & division.

Algorithm: Arithmetic operations

Step 1: Start
Step 2: [Read value of A & B]
Step 3: [Compute sum, difference, product & quotient]
Sum = A+B
diff = A-B
prod = A*B
quot = A/B
Step 4: [print the sum, diff, prod & quot]
print 'Sum of A & B =', sum
print 'Difference of A & B =', diff
print 'product of A & B =', prod
print 'Quotient of A & B =', quot
Step 5: [End of algorithm]
Stop

Flowchart

*This is a chart showing a flow of logic involved in solving a problem. This is defined for an algorithm.

* The flowchart can be defined as a diagramatic representation of an algorithm.

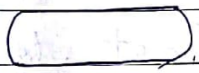
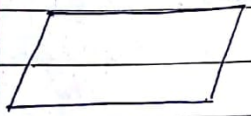
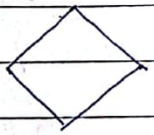
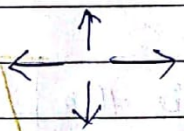
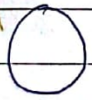
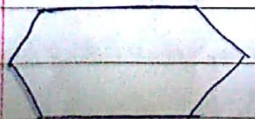
* It is also referred to as the blueprint of an algorithm.

3/10/2016

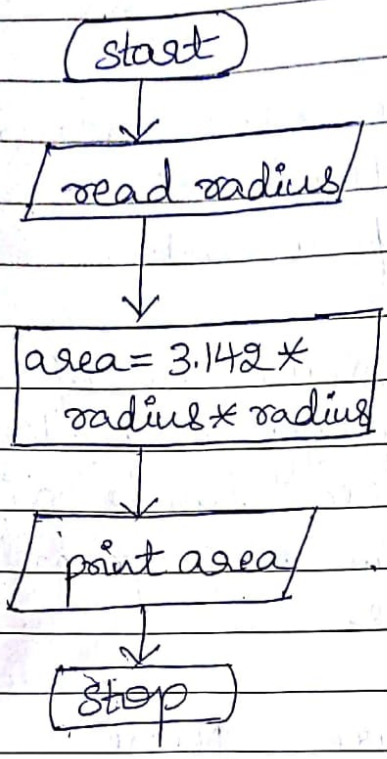
* The flowchart is an easy way to understand & analyze the problem. It is a useful aid for programmers & system analysts.

*Flowcharts are classified into 2 types.
1. Program flowcharts
2. System flowcharts.

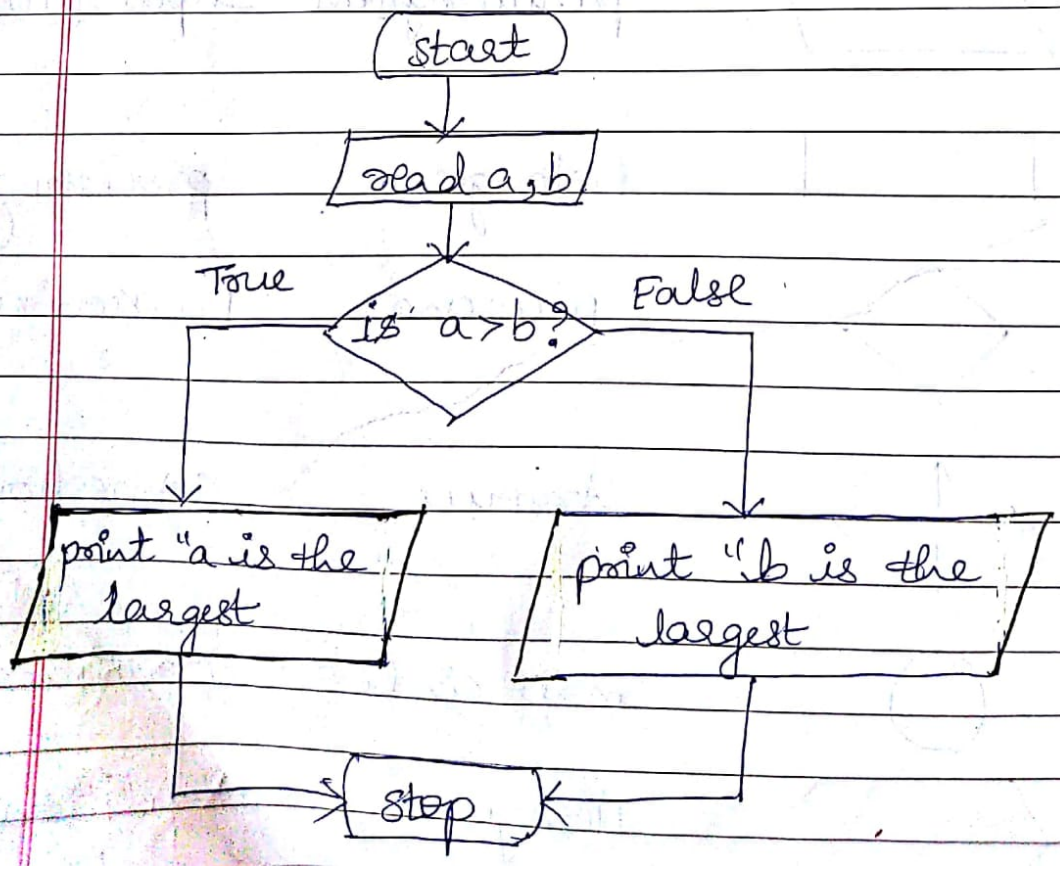
* Flowcharts make use of geometrical figures, to specify a particular operation. The below table shows different geometrical figures used in a program flowchart with their functions.

Geometrical figure	Name	Function
	oval	Start & Stop
	parallelogram	Input & output
	Rectangle	processing
	Diamond	Decision making
	Arrows	Connections
	Small circle	continuation
	Hexagon	Repetition / Looping

Example 1 : Draw a flowchart to find the area of circle when its radius is given.



Example 2 :- Draw a flowchart to find the largest of 2 numbers.



* Structure of C program :-

The complete structure of a C program is shown below

```
/* comment section */
preprocessor statements / Inclusion section
global declarations;
main()
{
    declarations; → (Local variable)
    statements;
}
user defined functions
```

* Preprocessor statements

These statements begin with # symbol & are also called the preprocessor directives. These statements direct the C preprocessor to include header files & also symbolic constants into a C program.

Some of the preprocessor statements are given below.

#include<stdio.h> : for std input/output functions

#define NULL 0 : for defining symbolic constant, NULL = 0

* Global Declarations

Variables or functions whose existence is known in the main() function & other user defined functions are called as the

global variables (of functions) & their declarations are called global declarations.

* The main() functions

* the execution of a C program starts with main(). No C program is executed without the main() function. It calls other library functions & user defined functions.
* The left brace '{' indicates the beginning of the main() function. The right brace indicates the end of the main() function.

* Declarations

It is a part of C program where all the variables, arrays, functions etc. used in the C program are declared & may be initialized with their basic data types.

* Statements

* These are instructions to the computer to perform some specific operations. They may be input-output statements, arithmetic statements, control statements. They also include comments. Comments are explanatory notes on some instructions.
* the statements to be commented must be enclosed within /* and */.

User-defined functions

* These are subprograms. It contains a set of statements to perform a specific task. These are written by the user, hence it is named as user-defined functions.

A sample C program is given below:

```
#include<stdio.h>
void main()
{
    printf("welcome to C\n");
}
```

Output
Welcome to C

* The 1st line tells the compiler to include the standard input/output header file to perform the reading & printing of data.

* The 2nd line is the main(),

* The body of C program contains only 1 statement i.e.,

printf("welcome to C\n");

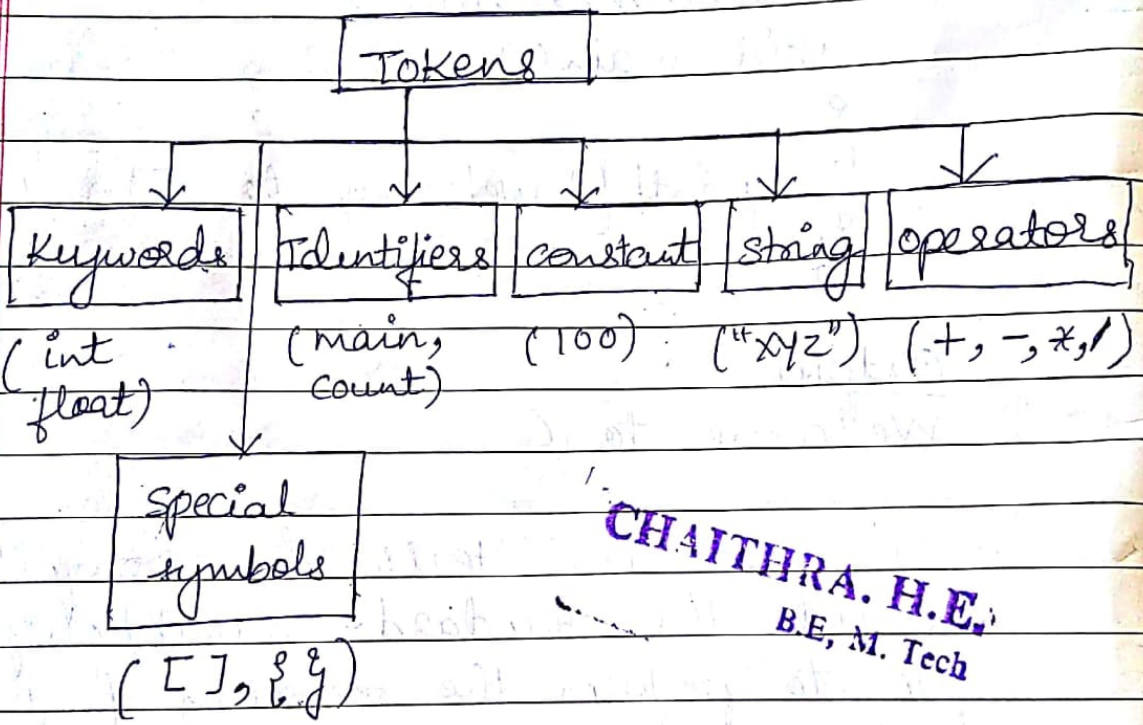
* When this statement is taken for execution, main() calls the printf()

function, which is included in <stdio.h>. This printf() prints the "Welcome to C" on the computer screen.

* C Tokens :-

It is the basic & the smallest units of a C program.

* There are 6 types of tokens in C.



* Keywords & Identifiers

* Each words in a C program is either a Keywords or an Identifier.
* All Keywords are the sequence of characters that have 1 or more fixed meanings.
* All Keywords should be written in lower letters.

Eg:-

auto	break	char	const	continue
double	else	if	float	int
void	while	switch	short	return

* Identifiers

* It is the name given to the program elements such as variables, array & functions.
* It is the sequences of alphabets & digits.

Rules for forming Identifier names

* The first character must be an alphabet (uppercase or lowercase) or an underscore.
* No special characters are allowed.
* No 2 successive underscores are allowed.
* Keywords should not be used as an identifiers.

* Datatypes :-

Datatypes indicate the type of data that the variable can hold. The data may be numeric or non-numeric in nature.

It is classified into -

1. Built-in (Basic) data type

2. Derived data type
3. user-defined data type

1. Built-in data type

i) Integer
ii) Floating point number
iii) Double precision
iv) Characters.

In C, there is one & only one non-specific data type called void. It does not specify anything.

Types	Keyword	Size (Bytes)
Integer	int	2
Floating-point	float	4
Double precision	double	8
Character	Char	1

non-specific void

i) int :- This is a keyword used to indicate an integer number

* Any integer number is a sequence of digits without decimal point.

* The range of int depends on the word length of computer.

* Word length is the number of bits that can be accessed at a time by the processor.

* The 16-bit computer handles the integers from -32,768 to +32,767.

ii) float :-

* This is a keywords used to indicate a floating-point number.

* It is same as the real numbers.

* It is called as the floating-point number because the decimal point is shifted either to the left or to the right of some digits during manipulation.

* The range of float is -3.4E+38 to +3.4E+38.

* It has 6 digits of precision.

iii) double :-

* This keyword is used to indicate the double-precision floating-point number.

* It stores the 16 significant digits after the decimal point.

Eg :- 234.0000000000000000

* The size of double is 8 bytes

* the range is from

+1.7E-308 to +1.7E+308.

iv) char:-

* This keyword is used to indicate the characters

*. The size of char is 1 byte

* The range is from -128 to +127

* A data may be a character constant or string constant.

* Character constant is a single character enclosed within a pair of apostrophe

Eg:- 'a', 'p' ---etc.,

* String constant is set of characters closed within a pair of double quotes

Eg:- "computer"

*Data type modifiers

The basic data types except void can be modified using a series of type modifiers to fit the requirement of a particular program more closely.

These modifiers are also called qualifiers. They are:

1. Signed
2. unsigned
3. short
4. Long

*Signed :- * It is applied to integer variables.
* It can be applied with a character data type to create small integer.

* Size of signed char is 1 byte

* The range is -128 to +127

* unsigned :-

* This can be used for both int and char. Range
* Size of unsigned int is 0 to 65535.
*. It is used to create a unsigned integer.

* Size of unsigned char is 1 byte

* Range is 0 to 255.

* long :-

* This is applied to int data type.

* Size of long int is 4 byte

Short:-

* This makes the size of an int half.

* Size of short int is 2 bytes

2. Derived data type:-

It includes

1. Pointer types
2. Array types
3. Structure types
4. Union types
5. Function types

* Variables

* The quantity that changes during the execution.

* The variables represent a particular memory location where the data can be stored.

* It is used to denote constants, functions, arrays etc.,

Eg:- sum, area, age, length etc.,

*Rules for forming variable names

1. The 1st character of a variable name must be an alphabet or an underscore (_).

2. All succeeding characters consists of letters & digits.

3. Keywords should not be used as variables.

4. Special characters are not allowed.

* Declaration of variables

* The purpose of declaring variables is to reserve the amount of memory required for these variables.

Syntax is

Data_type	varlist	semicolon

where,

Data_type → is a basic data type such as int, float, char or double.

varlist → one or more variables of type data_type. & these variables must be separated by commas.

semicolon → a delimiter of this declaration.

Eg:-
```
int length;
float area;
char ch;
int x,Y,z;
```

* Assigning values to variables

* The process of giving values to variables is called the assignment of values.

* The assignment operator '=' is used to assign a value to a variable.

* Syntax is

variable_name	=	value	;

where,

variable_name ⇒ the memory location where the 'value' should be stored.

= ⇒ assignment operator.

value ⇒ is a constant or a variable.

there are 2 methods :-

1. Initial values can be assigned to variables within the declaration section.

Eg :- int x = 1 ;
char ch = 'y' ;

2. Initial values are assigned to the variables in the executable part of a program.

Eg :- x = 10 ;
ch = 'y' ;

Programs on data types

1) Write a C program to print a message.

```
#include<stdio.h>
void main()
{
clrscr();
printf("Hello world");
getch();
}
```


2) Write a C program to display the message line by line.

```
#include<stdio.h>
void main()
{
clrscr();
printf("Hello \n");
printf("How are you");
getch();
}
```

3) write a C program to print the integer value.

```
#include<[redacted]stdio.h>
void main()
{
int a;
clrscr();
a=10;
```

```
printf("The a value is %d", a);
getch();
}
```

4) Write a C program to perform the addition of 2 integers.

```
#include<stdio.h>
void main()
{
int a, b, sum;
clrscr();
a=10;
b=20;
sum=a+b;
printf("The sum of a and b is %d", sum);
getch();
}
```


5) Write a C program to print the integers value.

```
#include<stdio.h>
void main()
{
int a, b;
clrscr();
a=10;
b=20;
printf("The value of a is %d and the value of b is %d", a, b);
getch();
}
```

6) Write a C program which accepts the integers & prints the integer's value.

```
#include<stdio.h>
void main()
{
  int a,b;
  clrscr();
  printf("Enter a value\n");
  scanf("%d",&a);
  printf("Enter b value\n");
  scanf("%d",&b);
  printf("The value of a is %d and b value is %d",a,b);
  getch();
}
```

7) Write a C program which accepts the integer & performs the addition of integers & print the sum.

```
#include<stdio.h>
void main()
{
  int a, b, sum=0;
  clrscr();
  printf("enter values of a and b\n");
  scanf("%d%d",&a,&b);
  sum=a+b;
  printf("the sum is %d",sum);
  getch();
}
```

8) Write a C program to find the area of circle.

Formula

area = πr^2

area = pi * r * r or pi * pow(r, 2)

program :-

```
#include<stdio.h>
#include<math.h>
#define pi 3.142

void main()
{
  float r, area;
  clrscr();
  printf("enter value of radius\n");
  scanf("%f", &r);
  area = pi*r*r;
  printf("area of circle is %f", area);
  getch();
}
```


9) Write a C program to find the area of rectangle.

Formula

area = b x h

program

```
#include<stdio.h>
#include<math.h>
void main()
{
    int b, h;
    float int area;
    clrscr();
    printf("enter the value of b and h");
    scanf("%d%d", &b, &h);
    area = b*h;
    printf("area of rectangle is %d", area);
    getch();
}
```

10) Write a C program to find the area of triangle.

```
#include<stdio.h>
#include<math.h>
void main()
{
    float a, b, c;
    float s, area;
    clrscr();
    printf("enter the value of a, b and c\n");
    scanf("%f%f%f", &a, &b, &c);
    s = (a+b+c)/2.0;
    area = sqrt(s*(s-a)*(s-b)*(s-c));
    printf("area=%f", area);
    getch();
}
```

11) Write a C program to accept character, integer & floating point number & display the same.

```
#include<stdio.h>
#include<conio.h>
void main()
{
char ch;
int a;
float b;
clrscr();
printf("enter a character\n");
scanf("%c",&ch);
printf("enter a number\n");
scanf("%d",&a);
printf("enter a floating point number\n");
scanf("%f",&b);
printf("character is %c",ch);
printf("integer is %d",a);
printf("Floating point number %f",b);
getch();
}
```

13) write a C program to swap 2 number using temporary variable.

```
#include<stdio.h>
void main()
{
  int a, b, temp;
  clrscr();
  printf("enter the value of a & b\n");
  scanf("%d%d", &a, &b);
  temp = a;
  a = b;
  b = temp;
  printf("After swapping a is %d\n b is %d
                                  a, b);

  getch();
}
```

14) Write a C program to swap 2 numbe without using temporary variable.

```
#include<stdio.h>
void main()
{
  int a, b;
  clrscr();
  printf("enter the value of a & b\n");
  scanf("%d%d", &a, &b);
  a = a+b;
  b = a-b;
```

```
a = a-b;
printf("after swapping a is %d\n
    b is %d", a, b);
getch();
}
```

Tracing a = 10 b = 5

a = a+b = 10+5 = 15

b = a-b = 15-5 = 10 → b

a = a-b = 15-10 = 5 → a

Finally a = 5 b = 10

Formatted Input & output Statements

- scanf()
- printf()

* Formatted I/O functions enable the user to specify the type of the data & the way in which it should be read in & written out.

* Formatted Input function :-

- scanf()

* scanf() :- It is a function used to read the data of any type such as integer, character, float, double & string from the keyboard.

Syntax :- scanf("format string", &var1, ---, &varn);

where,

- format string → consists of 1 or more format specifiers beginning with %.
- &var1, ---&varn → List of address of variables.

Rules :-

1. The format string should be enclosed within a pair of double quotes.

2. The address of variable should be separated by commas.

3. The values for the address-list variables should match in number, type & the order of the variable.

4. The address of var list should not be enclosed within double quotes.

Data type	Format specifier	Meaning
int	%d	Reads decimal integer value
	(long int) %ld	Reads long integer value
	%u	Reads unsigned integer value
	%hd	Reads short integer value
	%o	Reads octal value
	%x	Reads hexadecimal value
char	%c	Reads character value
	%s	Reads a string
float	%f	Reads floating point number
	%e	Reads floating point number in exponential.
double	%lf	Reads long floating point numbers or double.

* Integer input :-

* %d is used as a format specifier

* Eg:- scanf("%d", &number);

* If there are more than 1 variable format specifier within format string then specify them by zero blank space.

Eg:- scanf("%d%d%d", &a, &b, &c);

*We can accept data through the field specification.

* Field specification means number of columns or field reserved for the input.

Eg:- scanf("%3d", &num);

If the input data to be assigned to num is 2345, then only 234 is stored in memory location.

*Character input :-

%c is used as a format specifier.

Eg:- scanf("%c", &char);

* If character input is a name, then a sequence of characters that constitutes the name.

* Floating-point number Input :-

* %f is used as a format specifier.

eg:-
```
float a;
scanf("%f", &a);
```

* Mixed mode Input

*scanf() can accept more than one type of data (mixed mode).

* The number of input variables, data type & their order should be proper.

eg:-
```
int a;
float b;
char c;
scanf("%d%f%c", &a, &b, &c);
```

* Formatted output function :-

- printf() :-

* This function display the data on the screen.

Syntax :-

printf("format string", var1, ---- varn);

where,

- format string → consists of 1 or more format specifier.
- var1, var2;;varn represent the values of variable.

Data type	Format	Meaning
int	%d	To print decimal integer
	%o	To print octal value
	%u	To print unsigned integer
	%x	To print hexadecimal value
	%hd	To print short integer value
float	%f	To print floating pt number
	%e	To print floating pt number
char	%c	To print character value
	%s	To print string
double	%lf	To print double
long int	%ld	To print long integer value.

* Integer output :- Syntax :- %wd
where w → Indicates number of column
%d is used as format specifier.

Eg:- printf("%d", num);

Eg:- num = 3456;

printf("%4d", num);

Then output is pictorially shown as:-

3	4	5	6

if → num = 345;
printf("%4d", num);

output :

	3	4	5

num = 798
printf("%2d", num);
output :

7	9	8

* If the space specified is less than requirement then computer provides space to print.

* For +ve integers the fields will be allocated using right justification.
* For -ve integers → left justification.

Eg:- num = 345;
printf("%-5d", num);

output :

3	4	5		

*If you want to fill all the empty space by 0, then write

printf("%05d", num);

output :-

0	0	3	4	5

Example 1:- WAP to show the usage of formatted output function.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int N= 4567;
  clrscr();
  printf("%d\n", N);
  printf("%.5d\n",N);
  printf("%.-5d\n",N);
  printf("%.06d", N);
  getch();
}
```

CHAITHRA. H.E.
B.E, M. Tech

output :

4	5	6	7	
	4	5	6	7
4	5	6	7	
0	0	4	5	6	7

Example 2:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
  float n = 2.345;
  clrscr();
```

```
printf("%f\n", n);
printf("%8.2f\n", n);
printf("%.4.2f", n);
getch();
}
```

output :- 2.345000
2.35
2.35