

Programming for Problem Solving KCS 101

Unit 1st

Computers Fundamentals: Classification of Computers, Application of Computers, Basic organization of computer, Input and Output Devices, Computer memory, Computer Software. Operating systems

Algorithms and Programming Languages: Algorithm, Flowcharts, Pseudo code, Generation of Programming Languages.

Q1 What is Computers & its Advantages its classification & generations.

A computer is an electronic data processing device, which accepts and stores data input, processes the data input, and generates the output in a required format.

Functionalities of a Computer

If we look at it in a very broad sense, any digital computer carries out the following five functions –

Step 1 – Takes data as input.

Step 2 – Stores the data/instructions in its memory and uses them as required.

Step 3 – Processes the data and converts it into useful information.

Step 4 – Generates the output.

Step 5 – Controls all the above four steps.

Advantages of Computers

High Speed

- Computer is a very fast device.
- It is capable of performing calculation of very large amount of data.
- It can perform millions of calculations in a few seconds as compared to man who will spend many months to perform the same task.

Accuracy

- In addition to being very fast, computers are very accurate.
- The calculations are 100% error free.
- Computers perform all jobs with 100% accuracy provided that the input is correct.

Storage Capability

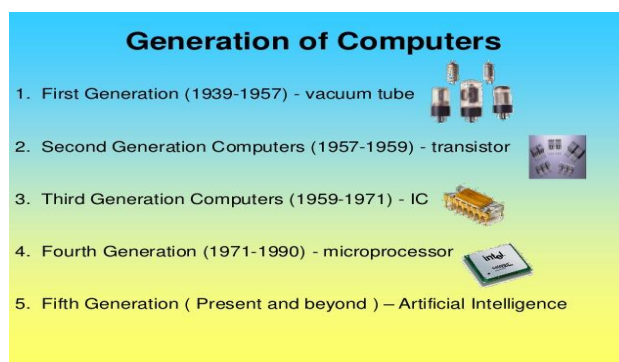
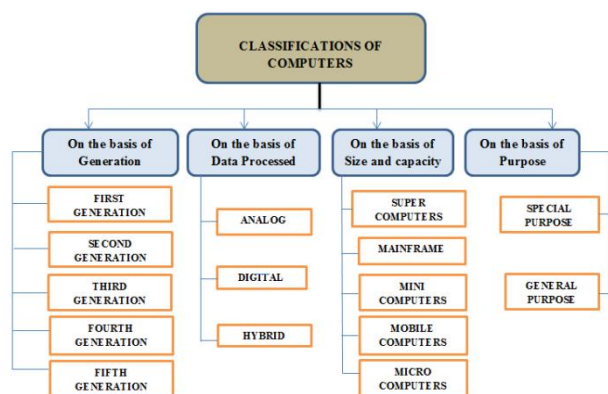
- Memory is a very important characteristic of computers.
- A computer has much more storage capacity than human beings.
- It can store large amount of data.
- It can store any type of data such as images, videos, text, audio, etc.

Versatility

- A computer is a very versatile machine.
- A computer is very flexible in performing the jobs to be done.
- This machine can be used to solve the problems related to various fields.
- At one instance, it may be solving a complex scientific problem and the very next moment it may be playing a card game.

Reliability

- A computer is a reliable machine.
- Modern electronic components have long lives.
- Computers are designed to make maintenance easy



Size	Characteristics	Uses
super computers	Largest, fastest, powerful and most expensive. they also generate a lot of heat	Used for advanced scientific research such as nuclear physics.
Main frames	Less powerful and less expensive than super computers. They also have a large storage capacity	Used to handle all kinds of problems whether scientific or commercial. I.e. performing complex mathematical calculations. they are mostly found in banks, hospitals, airports etc.
Mini computers	Smaller and less powerful than the main frame	Used in scientific laboratories, research institutions, engineering plants and places where processing automation is required.
Micro computers	Smallest, cheapest and relatively least powerful. Uses a micro processor to process data. Examples: desktop, laptop and personal digital assistant (PDA)	Used to perform a variety of tasks including research, communication, banking, learning institutions, libraries etc.

Q 2 The Components of Computer System, & Difference between Application & System Software

Hardware- Parts of the computer you can actually touch. (Monitors, keyboards, mouse, board, chips) Hardware (H/W), in the context of technology, refers to the physical elements that make up a computer or electronic system and everything else involved that is physically tangible.

Hardware components of a computer system:

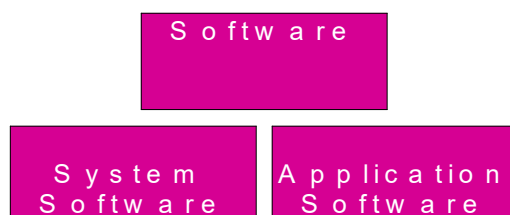
CPU - central processing unit : Makes decisions, performs computations, and delegates input/output requests.

Memory: Disk Drives, CD drives, Tape drives, USB flash drives. Stores information

Input devices: Keyboard, Mouse, Gets information from the user to the computer

Output devices: monitor : Sends information from computer to the user

Software - Programs that can be stored electronically. Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer. The two main categories of software are [application](#) software and [system software](#). An application is software that fulfills a specific need or performs tasks. System software is designed to run a computer's hardware and provides a platform for applications to run on top of.



System Software:

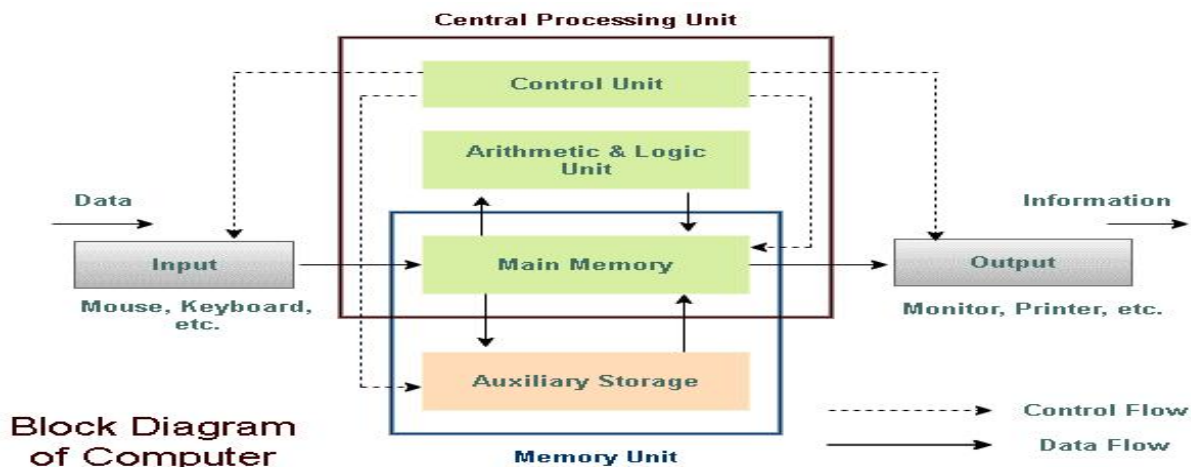
System Software is the type of software which is the interface between application software and system. Low level languages are used to write the system software. System Software maintains the system resources and gives the path for application software to run. An important thing is that without system software, system can not run. It is a general purpose software.

Application Software:

Application Software is the type of software that runs as per user request. It runs on the platform which is provided by system software. High level languages are used to write the application software. Its a specific purpose software.

S.No.	System Software	Application Software
1.	System software is used for operating computer hardware.	Application software is used by user to perform specific task.
2.	System softwares are installed on the computer when operating system is installed.	Application softwares are installed according to user's requirements.
3.	In general, the user does not interact with system software because it works in the background.	In general, the user interacts with application softwares.
4.	System software can run independently. It provides platform for running application softwares.	Application software can't run independently. They can't run without the presence of system software.
5.	Some examples of system softwares are compiler, assembler, debugger, driver, etc.	Some examples of application softwares are word processor, web browser, media player, etc.

Q3 Block Diagram of Computer , Neat Diagram of Digital Computer , Component of CPU



Input Unit

This unit contains devices with the help of which we enter data into the computer. This unit creates a link between the user and the computer. The input devices translate the information into a form understandable by the computer.

Output Unit

The output unit consists of devices with the help of which we get the information from the computer. This unit is a link between the computer and the users. Output devices translate the computer's output into a form understandable by the users.

Central Processing Unit (CPU) –

CPU is considered as the brain of the computer.

CPU performs all types of data processing operations.

It stores data, intermediate results, and instructions (program).

It controls the operation of all parts of the computer

CPU is considered as the brain of the computer. CPU performs all types of data processing operations. It stores data, intermediate results, and instructions (program). It controls the operation of all parts of the computer.

CPU itself has the following three components –

- ALU (Arithmetic Logic Unit) , Memory Unit
- Control Unit

ALU(Airthmatical & Logical Unit)

All the Arithmetical and logical operation in computer perform by ALU

This unit consists of two subsections namely,

Arithmetic Section & Logic Section

Arithmetic Section : Function of arithmetic section is to perform arithmetic operations like addition, subtraction, multiplication, and division. All complex operations are done by making repetitive use of the above operations.

Logic Section : Function of logic section is to perform logic operations such as comparing, selecting, matching, and merging of data

The Control Unit (CU)

This unit controls the operations of all parts of the computer but does not carry out any actual data processing operations.

Functions of control unit are –

It is responsible for controlling the transfer of data and instructions among other units of a computer.

It manages and coordinates all the units of the computer.

It obtains the instructions from the memory, interprets them, and directs the operation of the computer.

It communicates with Input/Output devices for transfer of data or results from storage.

It does not process or store data.

Memory

This unit can store instructions, data, and intermediate results. This unit supplies information to other units of the computer when needed. It is also known as internal storage unit or the main memory or the primary storage or Random Access Memory (RAM).

Its size affects speed, power, and capability. Primary memory and secondary memory are two types of memories in the computer.

Functions of the memory unit are –

It stores all the data and the instructions required for processing.

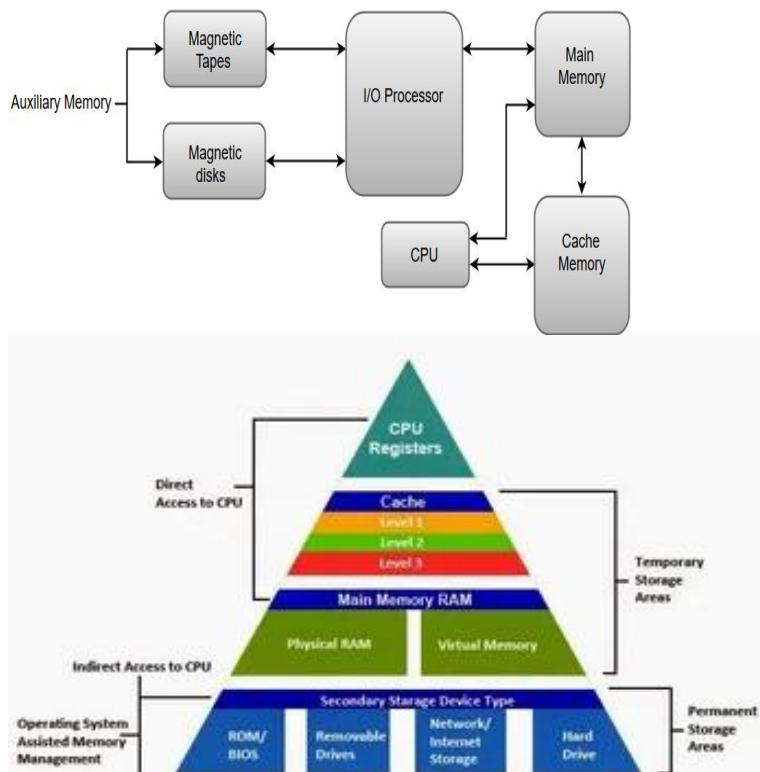
It stores intermediate results of processing.

It stores the final results of processing before these results are released to an output device.

All inputs and outputs are transmitted through the main memory.

Q 4 Different type of memory , Different type of storage , Difference between RAM & ROM , Memory Hierarchy

Memory Hierarchy in a Computer System:



Memory is the best essential element of a computer because computer can't perform simple tasks. The performance of computer mainly based on memory and CPU.

Memory is internal storage media of computer that has several names such as majorly categorized into two types, Main memory and Secondary memory.

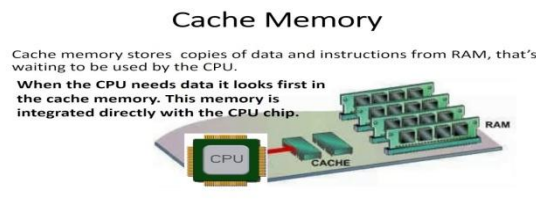
1. Main Memory /Primary Memory / Volatile Memory.
2. Secondary Memory/Auxiliary / Non Volatile Memory.

CPU Registers

Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as Processor registers.

A processor register may hold an instruction, a storage address, or any data (such as bit sequence or individual characters). The computer needs processor registers for manipulating data and a register for holding a memory address. The register holding the memory location is used to calculate the address of the next instruction after the execution of the current instruction is completed.

Cache Memory



Cache memory is a **very high speed semiconductor memory which can speed up the CPU**. It acts as a **buffer** between the CPU and the main memory. It is used **to hold those parts of data and program which are most frequently used by the CPU**. The parts of data and programs are **transferred from the disk to cache memory by the operating system**, from where the CPU can access them

The advantages of cache memory are as follows

Cache memory is faster than main memory.

It consumes less access time as compared to main memory.

It stores the program that can be executed within a short period of time.

It stores data for temporary use.

Main Memory

Main memory is divided into a large number of individual memory locations each, can hold a certain amount of data. each location has an address, which the CPU uses to designate which location to store data in or retrieve data from. (Can access info in billionths of a second).

Primary Memory also called as volatile memory because the memory can't store the data permanently. Primary memory select any part of memory when user want to save the data in memory but that may not be store permanently on that location. It also has another name i.e. RAM.

Two Categories Of Main Memory:

► RAM (RANDOM ACCESS MEMORY)

Read-write Memory & Contents Are Volatile

► ROM (READ ONLY MEMORY)

Read Only Memory - Permanent Storage, Can't Be Changed, Installed In Computer

Random Access Memory (RAM):

RAM (Random Access Memory) is the internal memory of the CPU for storing data, program, and program result. It is a read/write memory which stores data until the machine is working. As soon as the machine is switched off, data is erased. Access time in RAM is independent of the address, that is, each storage location inside the memory is as easy to reach as other locations and takes the same amount of time. Data in the RAM can be accessed randomly but it is very expensive.

RAM is volatile, i.e. data stored in it is lost when we switch off the computer or if there is a power failure. Hence, a backup Uninterruptible Power System (UPS) is often used with computers. RAM is small, both in terms of its physical size and in the amount of data it can hold.

- ▶ RAM is of two types –
- ▶ Static RAM (SRAM)
- ▶ Dynamic RAM (DRAM)

Static RAM (SRAM) : The word static indicates that the memory retains its contents as long as power is being supplied. However, data is lost when the power gets down due to volatile nature.

Dynamic RAM (DRAM) : DRAM, unlike SRAM, must be continually **refreshed** in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second.

Read Only Memory (ROM) :

ROM is permanent memory location that offer huge types of standards to save data. But it work with read only operation. No data lose happen whenever power failure occur during the ROM memory work in computers.

PROM , EPROM , EEPROM

ROM stands for **Read Only Memory**. The memory from which we can only read but cannot write on it. This type of memory is non-volatile. The information is stored permanently in such memories during manufacture.

A ROM stores such instructions that are required to start a computer. This operation is referred to as **bootstrap**. ROM chips are not only used in the computer but also in other electronic items like washing machine and microwave oven.

- ▶ ROM memory has several models such names are following.

1. PROM: Programmable Read Only Memory (PROM) maintains large storage media but can't offer the erase features in ROM. This type of RO maintains PROM chips to write data once and read many.

2. EPROM : Erasable Programmable Read Only Memory designed for recover the problems of PROM and ROM. Users can delete the data of EPROM thorough pass on ultraviolet light and it erases chip is reprogrammed.

3. EEPROM: Electrically Erasable Programmable Read Only Memory similar to the EPROM but it uses electrical beam for erase the data of ROM

	RAM	ROM
Volatility	RAM is volatile in nature as it automatically erased when computer shutdowns	ROM is non-volatile since it is never erased when there is any shutdown or restart of computer.
Accessibility	RAM can be directly accessed by the processor	ROM can't be directly accessed by the processor since it is transferred into RAM where it is executed by the processor.
Storage	RAM is used to store the temporary information for limited time.	ROM is used to store permanent information which can't be deleted.
Hardware structure	RAM is in form of chip while	ROM is generally optical drivers made of magnetic tapes
Cost	Costlier than ROM	Cheaper than RAM
Size	Chip Size is larger than ROM	Chip Size is smaller than ROM
Writing speed	Writing data to a RAM chip is a faster process	Writing data to a ROM chip is a slow process
Storage Limit	A RAM chip can store multiple gigabytes (GB) of data , up to 16 GB or more per chip	A ROM chip typically stores only several megabytes (MB) of data, up to 4 MB or more per chip
Examples	Static and dynamic RAM	PROM, EPROM and EEPROM are types of ROM.

Secondary Memory

This type of memory is also known as external memory or non-volatile. It is slower than the main memory. These are used for storing data/information permanently. CPU directly does not access these memories, instead they are accessed via input-output routines. The contents of secondary memories are first transferred to the main memory, and then the CPU can access it. For example, disk, CD-ROM, DVD, etc.

Characteristics of Secondary Memory

These are magnetic and optical memories.

It is known as the backup memory.

It is a non-volatile memory.

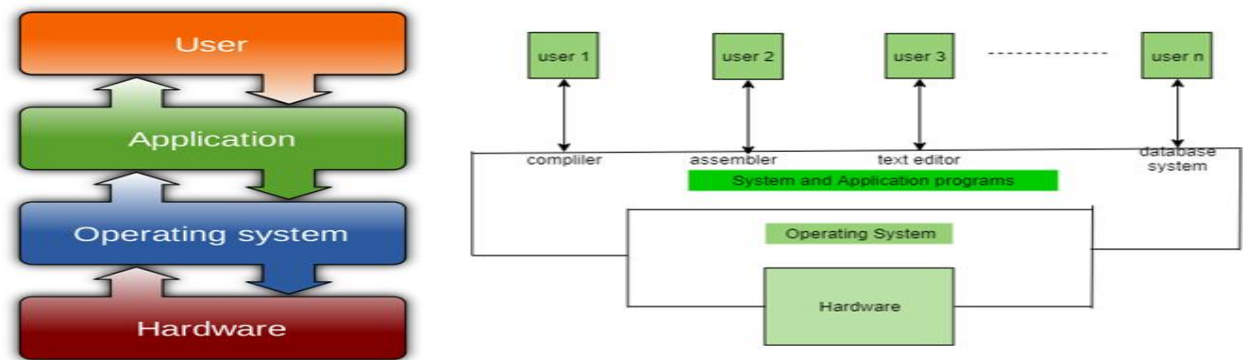
Data is permanently stored even if power is switched off.

It is used for storage of data in a computer.

Computer may run without the secondary memory.

Slower than primary memories.

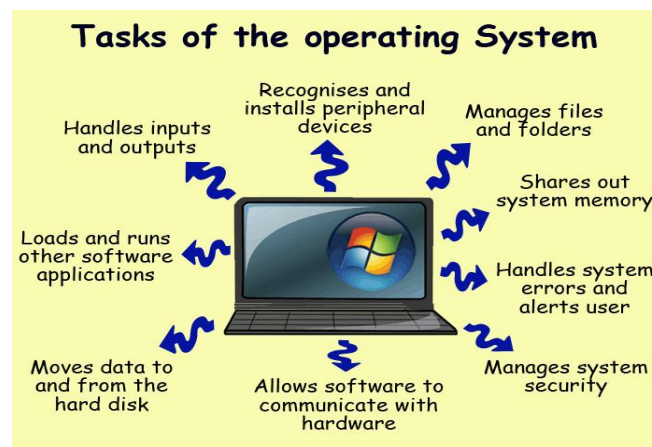
Q5 What is operating System / Different operating Systems/ Features/Services/Component of Operating System / Why OS work as a resource manager.



An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs



An Operating System provides services to both the users and to the programs. It provides programs an environment to execute. It provides users the services to execute the programs in a convenient manner.

Following are a few common services provided by an operating system –

1. Program execution

Operating systems handle many kinds of activities from user programs to system programs like printer spooler, name servers, file server, etc.

Each of these activities is encapsulated as a process.

Loads a program into memory. Executes the program.

Handles program's execution. Provides a mechanism for process synchronization. Provides a mechanism for process communication. Provides a mechanism for deadlock handling.

2. I/O Operation

An Operating System manages the communication between user and device drivers. Operating system provides the access to the required I/O device when required.

3. File system manipulation

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. Following are the major activities of an operating system with respect to file management –

Program needs to read a file or write a file. The operating system gives the permission to the program for operation on file. Permission varies from read-only, read-write, denied and so on.

Operating System provides an interface to the user to create/delete files. create/delete directories. AND create the backup of file system.

4...Error handling

Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware. Following are the major activities of an operating system with respect to error handling –

The OS constantly checks for possible errors. The OS takes an appropriate action to ensure correct and consistent computing.

5...Resource Management

In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management –

The OS manages all kinds of resources using schedulers. CPU scheduling algorithms are used for better utilization of CPU.

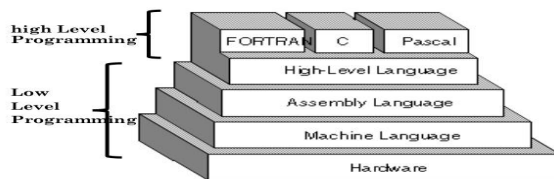
6...Protection

Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.

Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection –

The OS ensures that all access to system resources is controlled. The OS ensures that external I/O devices are protected from invalid access attempts. The OS provides authentication features for each user by means of passwords.

Q6 Different Programming Languages



A programming language is a **formal language comprising a set of strings that produce various kinds of machine code output**. Programming languages are one kind of computer language, and are used in computer programming to implement algorithms. These special languages are called *programming languages*.

These languages provide a way to write computer programs that are understood by both computers and people. Programming languages have their own vocabulary and rules of usage. Some languages are very technical, while others are similar to English.

Computer Programs

Software refers to programs that make the computer perform some task.

A *program* is a set of instructions that tells the computer what to do.

When you have written a program, the computer will behave exactly as you have instructed it. It will do no more or no less than what is contained in your specific instructions.

Machine Language?

A CPU is designed to “understand” a set of commands called an “*instruction set*” All instructions must be provided to the CPU in its native language, called *machine language*.

All data transmission, manipulation, storage, and retrieval is done by the machine using electrical pulses representing sequences of binary digits.

If eight-digit binary codes are used, there are 256 numbered instructions from 00000000 to 11111111

Instructions for adding two numbers would consist of a sequence of these eight-digit codes from 00000000 to 11111111.

Instructions written in this form are referred to as *machine language*. It is the native language that the CPU “speaks” and “understands”.

It is possible to write an entire program in machine language. However, this is very time consuming and difficult to read and understand.

Assembly Language

The programming language that is most like machine language is *assembly language*.

Assembly language uses letters and numbers to represent machine language instructions.

An *assembler* is a program that reads the codes the programmer writes in assembly language and “assembles” a machine language program based on those codes.

However, assembly language is still difficult to read.

- ▶ Assembly Code
- ▶ MOV A,[1] A=y
- ▶ MOV B,[2] B=z
- ▶ ADD A,B A=A+B;
- ▶ MOV [0],A x=A

High Level Languages

A *high-level language* is any programming language that uses words and symbols to make it relatively easy to read and write a computer program.

In a high-level language, instructions do not necessarily correspond one-to-one with the instruction set of the CPU. One command in a high-level language may correspond to **many** microprocessor instructions.

Many *high-level languages* have been developed. These include: FORTRAN, COBOL, BASIC, Logo, Pascal, C, C++, Java, Python, and others.

These languages simplify even further the terminology and symbolism necessary for directing the machine to perform various manipulations of data.

Advantages Of High Level Languages

Reduce the number of instructions that must be written.

Allow programs to be written in a shorter amount of time than a low-level language would take.

Reduce the number of errors that are made, because...The instructions are easier to read.

Are more portable (the programs are easier to move among computers with different microprocessors).

C Language code

```
Void main( )  
{  
    int i, j, k;  
    i = 3;  
    j = 2;  
    k = i + j;  
}
```

Q7 Different type of Translators & Difference between Compiler and Interpreter.

Translators are the converter that convert any type of programming Language in to Machine Code.

Set of programs used to develop software

Types of translators:

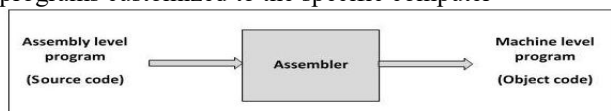
- Compiler
- Interpreter
- Assembler

Assembler

Assembler is a system software that converts assembly level programs to machine level code.

These are the advantages provided by assembly level programming –

Increases efficiency of the programmer as remembering mnemonics is easier , Productivity increases as number of errors decreases and hence debugging time , Programmer has access to hardware resources and hence has flexibility in writing programs customized to the specific computer



Compiler

System software that store the complete program, scan it, translate the complete program into object code and then creates an executable code is called a compiler. On the face of it compilers compare unfavorably with interpreters because they – are more complex than interpreters , need more memory space , take more time in compiling source code

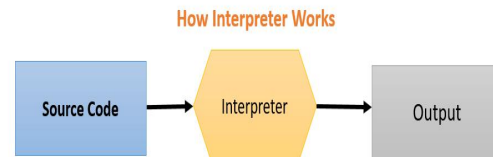
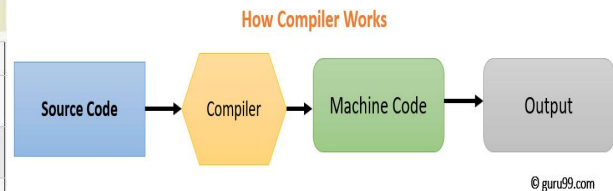
However, compiled programs execute very fast on computers.

Interpreter

The system software used to translate high level language source code into machine level language object code line by line is called an **interpreter**. An interpreter takes each line of code and converts it into machine code and stores it into the object file. The **advantage** of using an interpreter is that they are very easy to write and they do not require a large memory space. However, there is a major disadvantage in using interpreters, i.e., interpreted programs take a long time in executing. To overcome this **disadvantage**, especially for large programs, **compilers** were developed.

Difference between Compiler and Interpreter

No	Compiler	Interpreter
1	Compiler Takes Entire program as input	Interpreter Takes Single instruction as input
2	Intermediate Object Code is Generated	No Intermediate Object Code is Generated
3	Conditional Control Statements are Executes faster	Conditional Control Statements are Executes slower
4	Memory Requirement : More (Since Object Code is Generated)	Memory Requirement is Less
5	Program need not be compiled every time	Every time higher level program is converted into lower level program
6	Errors are displayed after entire program is checked	Errors are displayed for every instruction interpreted (if any)
7	Example : C Compiler	Example : BASIC



Q8 : Different Problem Solving Approaches

Pseudo Code Algorithms Flow Charts Program

Pseudo code : Pseudo code is a generic way of describing an algorithm without using any specific programming language-related notations. It is an outline of a program, written in a form, which can easily be converted into real programming statements. Pseudo code uses plain English statements rather than symbols, to represent the processes of a computer program.

It is also known as PDL (Program Design Language), as it emphasizes more on the design aspect of a computer program or structured English, because usually pseudo code instructions are written in normal English, but in a structured way. .

► Pseudo code to compute the area of a rectangle:

Get the length, l , and width, w

Compute the area = $l * w$

Display the area

► Pseudo code to check number is even or odd

Get a number

IF (number modulo 2) equals to 0

display number is even

ELSE

display number is odd

Algorithms

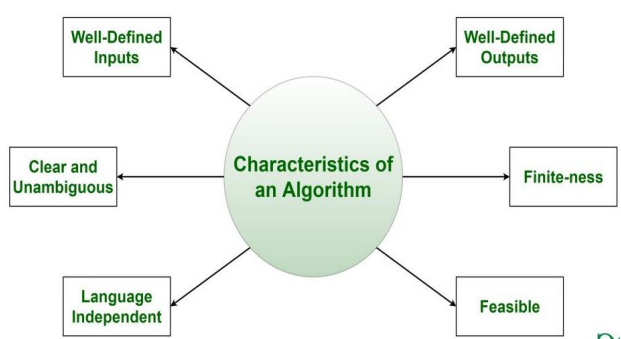
Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output.

A finite set of steps that must be followed to solve any problem is called an **algorithm**. Algorithm is generally developed before the actual coding is done.

It is written using English like language so that it is easily understandable even by non-programmers.

The word **Algorithm** means “a process or set of rules to be followed in calculations or other problem-solving operations”.

Therefore Algorithm refers to a set of rules/instructions that step-by-step define how a work is to be executed upon in order to get the expected results.

<p style="text-align: center;">Characteristics of an Algorithm</p> 	<p>How to Design an Algorithm?</p> <p>In order to write an algorithm, following things are needed as a pre-requisite:</p> <p>The problem that is to be solved by this algorithm.</p> <p>The constraints of the problem that must be considered while solving the problem.</p> <p>The input to be taken to solve the problem.</p> <p>The output to be expected when the problem the is solved.</p> <p>The solution to this problem, in the given constraints.</p>
<p>Examples Of Algorithms</p> <p>Write an algorithm to add two numbers entered by user.</p> <pre> Step 1: Start Step 2: Declare variables num1, num2 and sum. Step 3: Read values num1 and num2. Step 4: Add num1 and num2 and assign the result to sum. sum=num1+num2 Step 5: Display sum Step 6: Stop </pre>	<p>Problem – Design an algorithm to check numbers is even or odd.</p> <p>Step 1- Start</p> <p>Step 2- Read / input the number.</p> <p>Step 3- if $n \% 2 == 0$ then number is even.</p> <p>Step 4- else number is odd.</p> <p>Step 5- display the output.</p> <p>Step 6- Stop</p>

Flowchart

Flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing.

The process of drawing a flowchart for an algorithm is known as “flowcharting”.

A flowchart is a blueprint that pictorially represents the algorithm and its steps .

A Flowchart

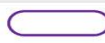





shows logic of an algorithm

emphasizes individual steps and their interconnections

control flow from one action to the next

Program flowcharts show the sequence of instructions in a single program or subroutine.

Different symbols are used to draw each type of flowchart.

Flowchart Symbols		
Here is a chart for some of the common symbols used in drawing flowcharts.		
Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.

Algorithm & Flowchart to find Area and Perimeter of Circle

R : Radius of Circle
 AREA : Area of Circle
 PERIMETER : Perimeter of Circle

Algorithm

Step-1 Start

Step-2 Input Radius of Circle say R

Step-3 $Area = 22.0/7.0 \times R \times R$

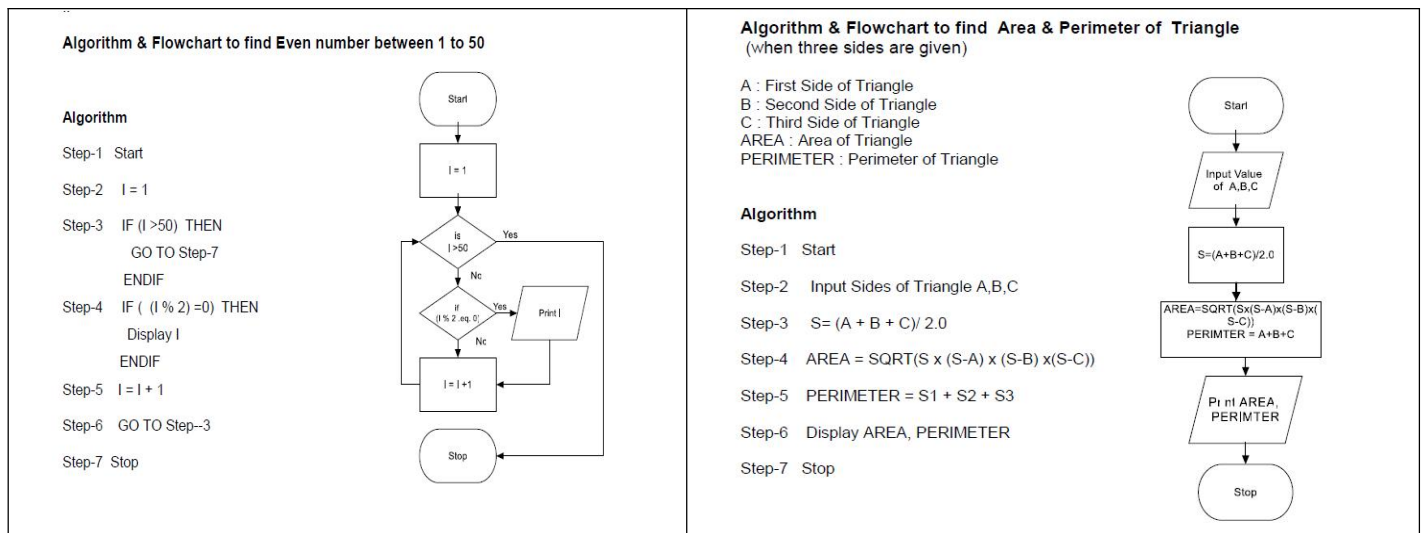
Step-4 $PERIMETER = 2 \times 22.0/7.0 \times R$

Step-5 Display AREA, PERIMETER

Step-6 Stop

```

graph TD
    Start([Start]) --> Input[/Input Value of R/]
    Input --> Area[AREA = 22.0/7.0 x R x R]
    Area --> Perimeter[PERIMETER = 2 x 22.0/7.0 x R]
    Perimeter --> Print[/Print AREA, PERIMETER/]
    Print --> Stop([Stop])
  
```



Programming for Problem Solving KCS 101

Unit 1

C Programming Language

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system.

C is the most widely used computer language.

C programming language some of the key advantages of learning C Programming:

Easy to learn

Structured language

It produces efficient programs

It can handle low-level activities

It can be compiled on a variety of computer platforms

C was invented to write an operating system called UNIX.

C is a successor of B language which was introduced around the early 1970s.

Today C is the most widely used and popular System Programming Language.

FEATURES OF C LANGUAGE

C is a structure oriented language: There is a predefined structure for each and every program, programmer have to follow the pre defined structure for a small program or a large program.

Self Defined Keyword: Keywords are the words whose meaning has already been explained to the C compiler. There are only 32 keywords available in C Language

Pure case Sensitive: C is pure case sensitive language if we write “printf ()” and “PRINTF ()” both are different .in C language all the predefined keywords and functions are in lowercase (small letters).

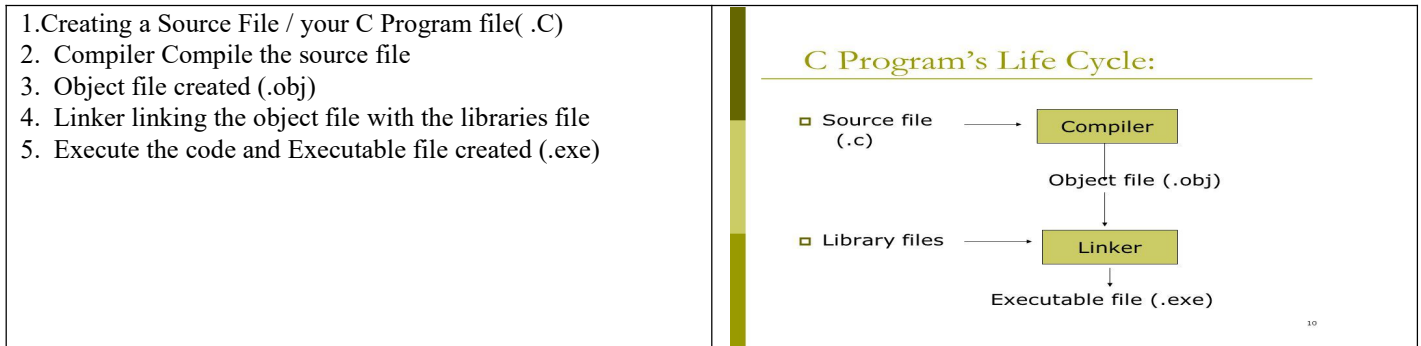
Highly Portable: C is a highly portable means that the c program written for one computer can be run on another computer without any modification. C program can be run on different operating system without any changes in the source code.

Robust Language: C is a robust language whose rich set of functions and keywords can be used to write any complex program. Robust language deals with errors by displaying a message and telling the user regarding the problem, usually asking them to re-type it, instead of just crashing. In C language the C Compiler checks out the error and displays the message for all the errors before execution of the program.

General purpose programming Language: C is a general purpose programming language it allows software developer to develop programs without knowing about the hardware platform where the program will be implemented.

Modularity: In C Language blocks of code can be stored in libraries for re-use in future programs. This concept of modularity also helps with C's portability and execution speed. These functions are instead stored in the C Standard Library where they can be called on when needed.

Q LIFE CYCLE OF C PROGRAM



Source File: The file written in C Programming Language by a programmer is called source file. The extension of this file is “.c”.

Compiler : Compiler check the error in our source code and if there is no any error in our source code then compiler convert source code into Machine code. So after the compilation, our source code is converted into object code (Machine code). There are many C compiler like Turbo c, Sub c ,Ritchie C compiler, SAS etc.

Object file: An object file is a file that contains the machine code that is usually not directly executable. Object files are produced by an assembler, compiler, or other language translator, and used as input to the linker.

Linker: linker is a computer program that takes one or more object files generated by the compiler and combine them into a single executable program. Linker provides the definition of all the keyword and function from C Library to machine at run time.

Executable File: Executable file hold the output of our program. An executable file is a file that is used to perform various functions or operations on a computer. Unlike a data file, an executable file cannot be read because it has been compiled. The .exe is the extension of the file .

PROGRAMMING LANGUAGE PROCESSORS

Editor: A program that allows text to be entered and changed.

Linker: A program which combines two or more object modules into a single object module or into an executable file . Linker is a program that takes one or more objects generated by a compiler and combines them into a single executable program.

Loader : Loader is the part of an operating system that is responsible for loading programs from executables (i.e., executable files) into memory, preparing them for execution and then executing them.

Q Structure of a C Program/ Why C is Structure Oriented Language.

#include <stdio.h>	Header File	include <stdio.h>
Void main()	Start Function	void main()
{		{
Variable Declaration;		int a ,b, c;
Input;		a=10;
Process;		b=20;
Output;		c=a+b;
}		printf(“%d “,c);
		}

preprocessor statements : These statement start with # symbol, and # is also called preprocessor directive .These statements direct the C preprocessor to include header file and also symbolic constant in to a C program.

include <stdio.h> : header file for standard input/output

include <conio.h> : header file for console

main() Function : This is the main function of a C program. Execution of a C program starts with main(). There is a single main function in a C program .because C is a case sensitive language so main() function written in lower case letters. The main() is a part of every C program we can write main() function in different ways.

Void main() :The function does not return any information to the operating system

int main() : The function should return integer value to operating system .when it is used the last statement must be return integer value means return(0).

Braces : Every C program uses a pair of curly braces { }. Its indicated the body of statements. The braces can also be used to indicate the beginning and end of user defined functions and the compound statements.

Declaration: The Declaration is the part of the C Program where all the variables, arrays, functions, constants etc are declare and may be initialized with their basic data types.

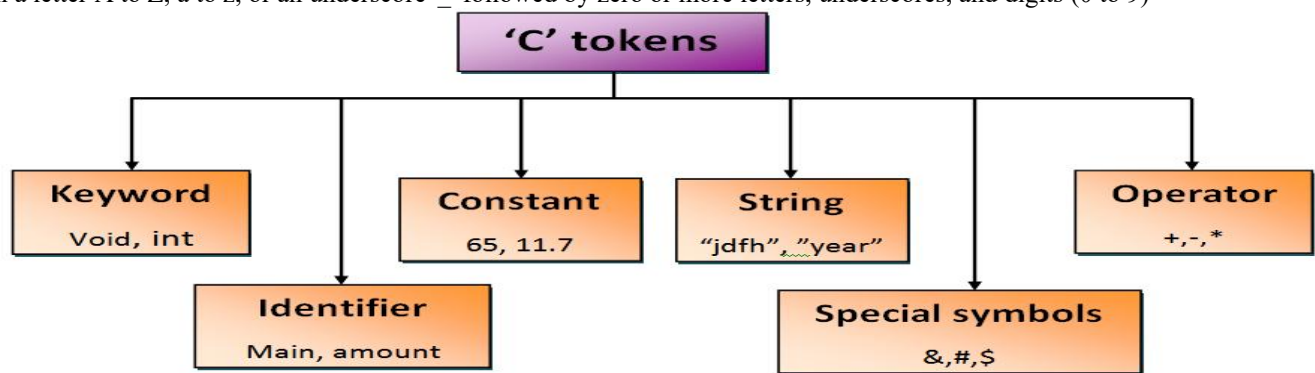
Process: These are instruction to the computer to perform some specific operations, they may be input output statements, arithmetic statement, control statements, and other that are used in the program for processing

Terminator “ ; ” semicolon : “ ; “ are used after each line, it indicate that the line is complete here and new line is start after semicolon.

Q Tokens in C Language

A C program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.

Identifiers : A C identifier is a name used to identify a variable, function, or any other user-defined item. An identifier starts with a letter A to Z, a to z, or an underscore '_' followed by zero or more letters, underscores, and digits (0 to 9)



Q Keywords

Keywords are the reserved words by the language for special use.

Every keyword has a special meaning.

Keywords are some reserved word which are not used as an identifier by the user in the program .

There are 32 keywords in C Language .

following are the keywords Auto,else,long,switch,break,enum,register,case, return,union,char,float,short,const,for,void, continue,goto,sizeof,default,if,static,while,do,intstruct

Q Variables (Identifiers)

variables are the names given to identify the specific program elements .

variable are also called an identifier that hold some value for processing.

Value of a variable can be change in the program.

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

They are used to denote constants, functions, arrays, name of files etc .

Rules for variable naming

The first character of a variable name must be an alphabet or an underscore(_)

All succeeding character consists of letters and digits.

Writing the variable name in lower case.

Keyword should not used as variable

Special character are not allowed

There is no limit on the number of character in the variable name

Always choose an appropriate variable name that makes proper sense to the user.

Blank spaces are not allowed as part of variable name

Some Valid variable Names:

San
number1
San_1
[san
num [20]
Area_of()
TOTAL_MARKS

Some Invalid Variable Names:

2san
Number 1
gross-salary
Cube's volume
\$total
For
number total

Declaration of Variables

All variable must be declared before they are used in C program. The purpose of declaring variables is to reserve the amount of memory required for these variables.

For declare any variable

► **Data type Variable Name**

int san ; Here san is a variable name that hold integer type value for processing and after the declaration of the variable its hold 2 byte of size in the Secondary memory

float Area; char ch; double density ;

int x,y,z; char name [20]; int a[10];

When a variable is declare an address of that variable is allocated in computer memory and value of that variable stored on that memory location.

Assign the value to variables:

The variable represents some memory location ,where the data is stored. Each variable is Associated with one or more values. The Assignment operator = is used to assign a value of a variable .

► Variable Name = value ;

int x = 1; float sum = 0.0 ; char ch = 's' ;

char ch [20] ="yash "

Q Constants :

The constants are the identifiers which do not change during the Execution of a program. Programming language C allow declaring constants variables, where value cannot be changed.

For declare a variable as a constant we used "const" keyword before the variable name;

const int a = 10;

int const x = 20;

float const pi = 3.14;

char const ch = 'x';

Q HEADER FILE\

- Collection of pre defined functions in c library.
- Header file is a file having syntax of library functions for common use & categorized in different set of files with .h extension.
- The header File are used to provide the necessary information in support of the various library functions.
- Each header file contains declaration for certain related library functions

Header file

Uses and functions

stdio.h

All Input/Output Function
printf(),scanf(),gets(),puts(),getchar(),putchar(),fgets(),fputs()

math.h

All Mathematical Functions
abs(),pow(),sqrt(),sin(),cos(),exp(),log()

string.h

All String Manipulation Functions
strcpy(),strlen(),strrev(),strcat(),strcmp(),strset(),strchr()

conio.h

All function related to the console
clrscr(),getch(),cprintf(),cputs(),cgets(),cscanf()

stdlib.h

Standard Library functions
exit(),max(),min(),malloc(),calloc(),realloc(),free()

ctype.h

Character conversion functions
isalpha(),isascii(),isdigit(),islower(),isupper(),tolower(),toupper()

Q Input /Output Function in C

For perform input output operation C provide standard input output library . The stdio.h header file containing all the input output functions.

Common input output functions are

printf() : used for common output

scanf() : used for common Input

getchar() : used for character input

putchar() : used for character output

gets() : used for string input

puts() :used for string output

Q Format Specifiers

printf() and scanf() function used some **format specifiers** for input and output of any type of value .

The **Format specifiers** tells the compiler that the value of the variable should be read or write in what format. The list of format specifiers is

printf() function used to display data on the standard output screen(Console) .The general form of printf() function is

Format Specifiers	Meaning
%c	For input and output a character value
%d	For input and output a integer value
%f	For input and output a float value
%i	For input and output a character
%s	For input and output a string value
%lf	For input and output a Double value
%ld	For input and output a Long value

Format of printf () function

```
printf("write message here that u want to display");  
printf ( " format specifier" ,variable name);  
printf ( "message = format specifier" , variable name);
```

format specifier specifies the type and format of the value to be displayed .

Some example of printf () function are:

```
printf ("%d",num);
```

it display the value that is hold by the variable num and the display value should be integer type

```
printf ("%f %d" ,a , b)
```

it display the value of a, b on the screen value of a is float type and value of b is integer type

scanf () function :

The scanf() function used to read the values for the variable in a C Program from the Keyboard. The scanf () function used to enter the numeric, character ,decimal and string type of data.

scanf (" format specifier " , &variable)

The format specifier specifies the type of the values which are to be transfer to the variable.

&variable : it specify the address of memory location where the values of input variable should be stored.

Some example of scanf () function are:

- ▶ scanf ("%d",&x);
- ▶ scanf ("%d %d ", &x , &y);
- ▶ scanf ("%d%f",&a, &x)

Q Backslash character Constants (Escape sequences)

A backslash constant is a combination of two characters backslash (\) and a character. The backslash constants are used in the output function. Backslash constant are also called Escape Sequence. A list of backslash constant is

Backslash Constant	Meaning
'\a'	System Alarm (bell)
'\b'	Back space
'\n'	New Line
'\t'	Horizontal Tab
'\v'	Vertical Tab
'\"'	Double quote
'\''	Single quote

Q DATA TYPE (primitive and non primitive)

As its name indicates, a data type represents a type of the data which you can process using your computer program. It can be numeric, alphanumeric, decimal, etc.

What type of data we used in our program. Data type indicates what type of value is hold be variable in a program. The data can be any type it may be numeric or non numeric .data type are used to define a variable before its use.

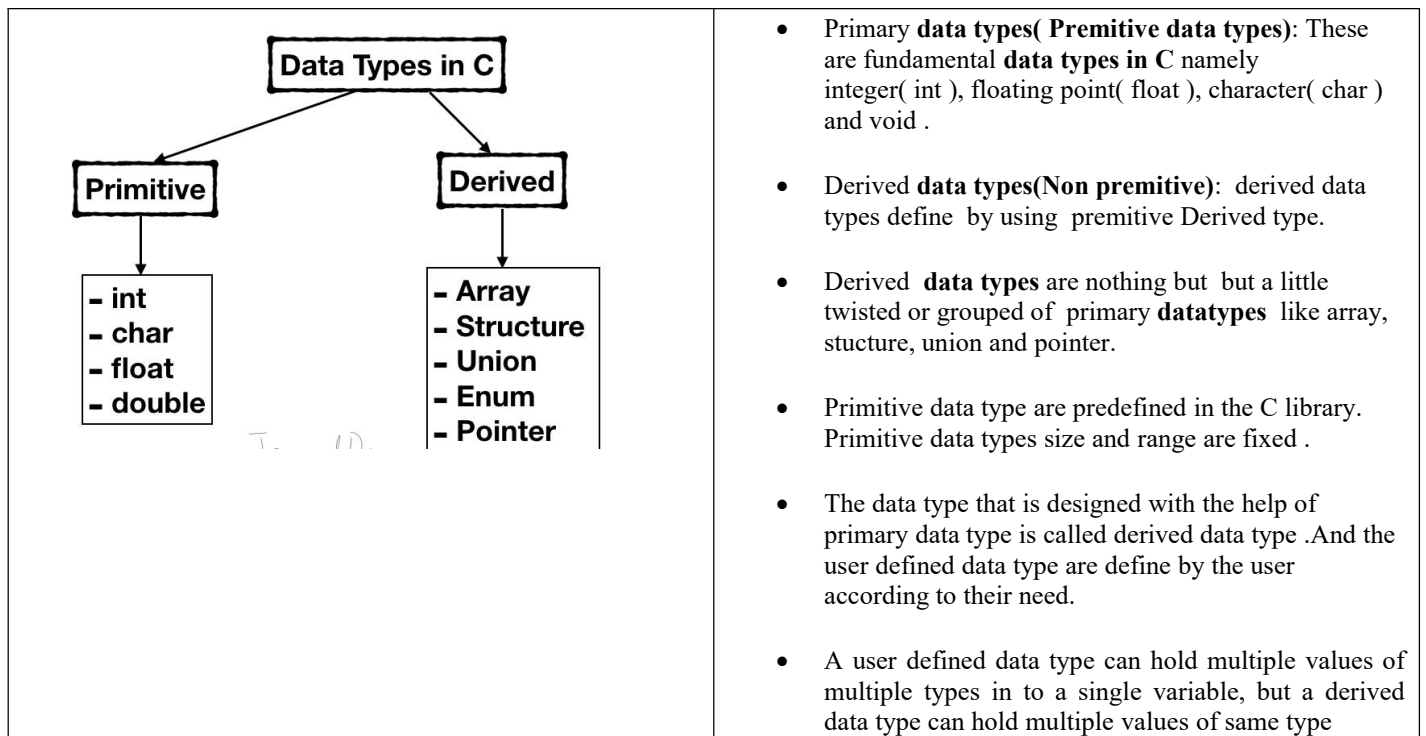
The data type determines the types of values and the range of values that can be stored in a variable.

In a class room there are two persons one is a girl and another is a boy whose name are same so we can identify them by

- ▶ Mr. XYZ
- ▶ Ms. XYZ

Here Mr. and Ms. Identify whether he is a boy or she is a girl.

So Data type is an identifier that shows the identification of data. What type of data we used in our program. Data type indicates the data that a variable can hold.\



Keyword	Size	Data Range	Format Specifier
char	1 Byte	-128 to +127	%c
unsigned char	8 Bytes	0 to 255	<-- -- >
int	2 Bytes	-32768 to +32767	%d
long int	4 Bytes	-2 ³¹ to +2 ³¹	%ld
unsigned int	2 Bytes	0 to 65535	%u
float	4 Bytes	-3.4e ³⁸ to +3.4e ³⁸	%f
double	8 Bytes	-1.7e ³⁸ to +1.7e ³⁸	%lf
long double	12-16 Bytes	-3.4e ³⁸ to +3.4e ³⁸	%Lf

Q Storage Classes

Storage class define the properties of a variable.

A storage class defines the scope (visibility) ,by default value ,storage location and life-time of variables and/or functions within a C Program.

'Storage' refers to the scope of a variable and memory allocated by compiler to store that variable. Scope of a variable is the boundary within which a variable can be used.

In C the variable is declared by the type of data they can hold .The variable is associated with the memory location within the computer where the value is stored. These variables may be stored in the register or main memory of the computer.

To indicate where the variable would be stored

how long they would exist

what would be the default value, is also define by the storage class.

C provide four storage classes:

- **auto**
- **static**
- **Register**
- **Extern**

Auto: auto is the default storage class .

Variable used as a local variable.

Initial value of variable is garbage without initialization Means if we not initialize any value to any variable the auto or default variable show the garbage value .

Memory is allocated automatically at the entry to a block and free automatically at exist from the block.

These variable can access only within the block in which they are declare.

```
Void main ()
{
    auto int j;
    Printf ("%d",j);
}
```

Output: garbage value

static :

The static class provides a lifetime to the variable over the entire program.

These variables are declared using static keyword.

Static variables do not get deleted when they fall out of scope, they are permanent and retain their value between call to the function.

The default value of static variable is zero if we declare a variable static and we cannot initialize that variable its value is zero.

```
Void main ()
```

```
{
static int j;
Printf ("%d",j);
}
```

Output: 0

Register :

Register is used to define the local variable that should be stored in a register instead of RAM.

This means the variable has a maximum size equal to the register size .

A value stored in a Register can be accessed faster than the one which is stored in the memory. Thus register variable provide certain control over efficiency of program execution. Generally those variable which are used repeatedly or is used at many places in a program or whose access time is critical may be declared to be register storage class.

- register int x;

Extern :

When we declare a variable extern our program does not actually reserve any memory for it ,extern means that the variable already exists externally to the function or file.

C provide extern storage class to declare any variable as a global variable.

extern variable are declared in the same way any other variable are declared.

They may be declared outside any function block. Memory is allocated to these variables when the program begins execution and the lifetime is until the program terminates .

```
extern int i=10; //extern variable
void main()
{
    printf ("%d",i);
}
```

Output is : 10

```
extern int i; //extern variable
void main()
{
    printf ("%d",i);
}
```

Output: Compilation error,
undefined symbol i.

Properties Storage Class	Storage	Default Initial Value	Scope	Life
Automatic	Memory	Garbage Value	Local to the block in which the variable is defined	Till the control remains within the block in which the variable is defined
Register	CPU Registers	Garbage Value	Local to the block in which the variable is defined	Till the control remains within the block in which the variable is defined
Static	Memory	Zero	Local to the block in which the variable is defined	Value of the variable continues to exist between different function calls
External	Memory	Zero	Global	Till the program's execution doesn't come to an end

WAP to find the root of Quadratic Equation

3 Write a program to Find the Area of a Triangle where a,b,c are sides

Formula Area = $\sqrt{s(s-a)(s-b)(s-c)}$ and $s=(a+b+c)/2$;

```
#include<stdio.h>
#include<math.h>
void main()
{
    int a,b,c;
    float s, area;
    printf("Enter the sides of a triangle:");
    scanf("%d%d%d",&a,&b,&c);
    s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("The Area is =%f",area);
}
```

Here we used math.h header file because we want to find the Square of any number which is mathematical function

Output: Enter the sides of a triangle : 5 6 4
The Area is = 4.96

```
main()
{
    int a,b,c,D;
    float x,y;
    printf("Enter coefficient of x^2, x and constant term");
    scanf("%d %d %d",&a,&b,&c);
    D=b*b-4*a*c;

    if(D<0)
        printf("Both roots are imaginary");
    if(D==0){
        printf("Both roots are equal");
        x=-b/(2*a);
        printf("Root is %f",x);
    }
    if(D>0){
        printf("Roots are real and distinct");
        x=(-b+sqrt(D))/(2*a);
        y=(-b-sqrt(D))/(2*a);
        printf("\nRoots are: %f ,%f",x,y);
    }
}
```