

# LECTURER NOTES

ON

## COMPUTER APPLICATION

FOR 1<sup>st</sup> and 2<sup>nd</sup> SEMESTER

Prepared By  
**ER. MADHUSMITA PATRA**  
Lecturer in Computer Application



**DEPARTMENT OF BASIC SCIENCE**

**BHARAT INSTITUTE OF ENGG & TECHNOLOGY**

(APPROVED BY AICTE & AFFILIATED TO SCTE & VT)

Sivaram Vihar, Ghatakeswer Hills, Mohada, Berhampur,  
Odisha. PIN: 760002

<http://www.biet.edu>

# Syllabus

## **Th.1b.COMPUTERAPPLICATION (1<sup>st</sup>/2<sup>nd</sup>sem Common)**

### **1. COMPUTER ORGANISATION**

Introduction to Computer Evolution of Computers Generation of Computers Classification of Computers Basic Organisation of Computer (Functional Block diagram) Input Devices, CPU & Output Devices. Computer Memory and Classification of Memory

### **2. COMPUTERS SOFTWARE**

Software concept, System software, Application software Overview of Operating System Objectives and Functions of O.S.,  
Types of Operating System: Batch Processing, Multiprogramming, Time Sharing OS Features of DOS, Windows and UNIX  
Programming Languages Compiler, interpreter Computer Virus Different Types of computer virus  
Detection and prevention of Virus Application of computers in different Domain

### **3. COMPUTER NETWORK AND INTERNET**

Networking concept, Protocol, Connecting Media, Data Transmission mode Network Topologies, Types of Network  
Networking Devices like Hub, Repeater, Switch, Bridge, Router, Gateway & NIC  
Internet Services like E-Mail, WWW, FTP, Chatting, Internet Conferencing, Electronic Newspaper & Online Shopping  
Different types of Internet connectivity and ISP

### **4. FILE MANAGEMENT AND DATA PROCESSING**

Concept of File and Folder  
File Access and Storage methods. Sequential, Direct, ISAM Data Capture, Data storage  
Data Processing and Retrieval

### **5. PROBLEM SOLVING METHODOLOGY**

Algorithm, Pseudocode and Flowchart Generation of Programming Languages Structured Programming Language  
Examples of Problem solving through Flowchart

### **6. OVERVIEW OF C PROGRAMMING LANGUAGE**

Constants, Variables and Data types in C Managing Input and Output operations. Operators, Expressions, Type conversion & Type casting  
Decision Control and Looping Statements (If, If-else, If-else-if, Switch, While, Do-while, For, Break, Continue & Goto)  
Programming Assignments using the above features.

### **7. ADVANCED FEATURES OF C**

Functions and Passing Parameters to the Function (Call by Value and Call by Reference) Scope of Variables and Storage Classes  
Recursion Function and Types of Recursion  
One Dimensional Array and Multidimensional Array String Operations and Pointers  
Pointer Expression and Pointer Arithmetic Programming Assignments using the above features. Structure and Union (Only concepts, No Programming)

# Contents

<b>Sl.No.</b>	<b>ChapterName</b>
1	ComputerOrganization
2	ComputerSoftware
3	ComputerNetworkandInternet
4	FileManagementandDataProcessing
5	ProblemSolvingMethodology
6	Overview of C ProgrammingLanguage
7	AdvancedFeatures of C
8	Chapter-wiseMultipleChoiceQuestions
9	References

# CHAPTER-1:COMPUTERORGANISATION

## What is computer?

Computer is a electronic machine through which we input the data ;manipulate or process it then produce a output and finally display in computer screen.

## What is data?

Data is nothing but it is raw facts and figures which present in the real world.

Ex:-pen, fan, studentetc.

## What is information?

Information is nothing but it is a collection of data which has some meaning.

Ex:-I am reading.

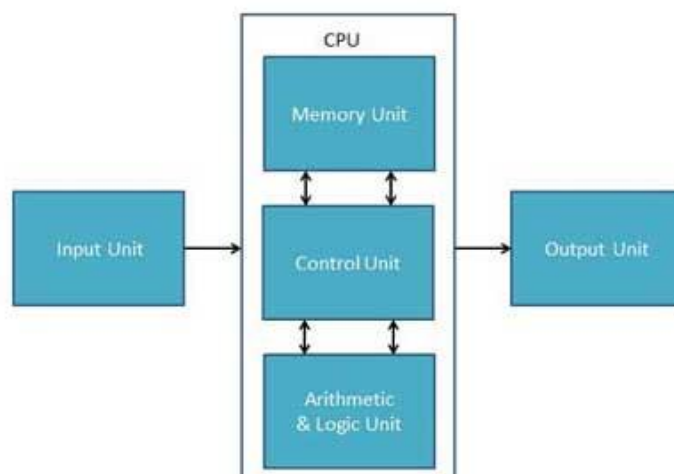
Here “I” is data and “am” and “reading” both are also data .when it combines, ithas a meaning.

## COMPUTER ORGATION:-

### BASIC OF COMPUTER ORGANISATION:-

All types of computers follow the same basic logical structure and perform the following five basic operations for converting raw input data into information useful to their users.

S.No.	Operation	Description
1	Take Input	The process of entering data and instructions into the computer system.
2	Store Data	Saving data and instructions so that they are available for processing as and when required.
3	Processing Data	Performing arithmetic, and logical operations on data in order to convert them into useful information.
4	Output Information	The process of producing useful information or results for the user, such as a printed report or visual display.
5	Control the workflow	Directs the manner and sequence in which all of the above operations are performed.



### < FUNCTIONAL BLOCK DIAGRAM >

#### 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.

### **Example of Input device:-**

Following are some of the important input devices which are used in a computer –

- Keyboard
- Mouse
- Joy Stick
- Light pen
- Track Ball
- Scanner
- Graphic Tablet
- Microphone
- Magnetic Ink Card Reader(MICR)
- Optical Character Reader(OCR)
- Bar Code Reader
- Optical Mark Reader(OMR)

### **Keyboard:-**

Keyboard is the most common and very popular input device which helps to input data to the computer. The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing additional functions.

Keyboards are of two sizes 84 keys or 101/102 keys, but now keyboards with 104 keys or 108 keys are also available for Windows and Internet.

### **Mouse:-**

Generally, it has two buttons called the left and the right button and a wheel is present between the buttons. A mouse can be used to control the position of the cursor on the screen, but it cannot be used to enter text into the computer.

### **Joystick:-**

Joystick is also a pointing device, which is used to move the cursor position on a monitor screen. It is a stick having a spherical ball at its both lower and upper ends. The lower spherical ball moves in a socket. The joystick can be moved in all four directions.

### **Light Pen:-**

Light pen is a pointing device similar to a pen. It is used to select a displayed menu item or draw pictures on the monitor screen. It consists of a photocell and an optical system placed in a small tube.

### **CPU (Central Processing Unit):-**

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.

Central Processing Unit (CPU) consists of the following features –

- 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 following three components.

- Memory or Storage Unit
- Control Unit
- ALU(Arithmetic Logic Unit)

### **Memory or Storage Unit:-**

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.

### **Control Unit:-**

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

Functions of this 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.

### **ALU (Arithmetic Logic Unit):-**

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.

#### **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.

#### **Example of Output device:-**

Following are some of the important output devices used in a computer.

- Monitors
- Graphic Plotter
- Printer

## **Monitors**

Monitors, commonly called as **Visual Display Unit** (VDU), are the main output device of a computer. It forms images from tiny dots, called pixels that are arranged in a rectangular form. The sharpness of the image depends upon the number of pixels.

There are two kinds of viewing screen used for monitors.

- Cathode-Ray Tube (CRT)
- Flat-Panel Display

## **Printers**

Printer is an output device, which is used to print information on paper.

There are two types of printers –

- Impact Printers
- Non-Impact Printers

### **Impact Printers**

Impact printers print the characters by striking them on the ribbon, which is then pressed on the paper.

Characteristics of Impact Printers are the following –

- Very low consumable costs
- Very noisy
- Useful for bulk printing due to low cost
- There is physical contact with the paper to produce an image

These printers are of two types –

- Character printers
- Line printers

### **Character Printers**

Character printers are the printers which print one character at a time.

These are further divided into two types:

- Dot Matrix Printer(DMP)
- Daisy Wheel

### **Dot Matrix Printer**

In the market, one of the most popular printers is Dot Matrix Printer. These printers are popular because of their ease of printing and economical price. Each character printed is in the form of pattern of dots and head consists of a Matrix of Pins of size (5\*7, 7\*9, 9\*7 or 9\*9) which come out to form a character which is why it is called Dot Matrix Printer.

### **Advantages**

- Inexpensive
- Widely Used
- Other language characters can be printed

### **Disadvantages**

- Slow Speed
- Poor Quality

### **Daisy Wheel**

Head is lying on a wheel and pins corresponding to characters are like petals of Daisy (flower) which is why it is called Daisy Wheel Printer.

#### **Advantages**

- More reliable than DMP
- Better quality
- Fonts of character can be easily changed

#### **Disadvantages**

- Slower than DMP
- Noisy
- More expensive than DMP

### **Line Printers**

Line printers are the printers which print one line at a time.

These are of two types –

- Drum Printer
- Chain Printer

#### **Drum Printer**

This printer is like a drum in shape hence it is called drum printer. The surface of the drum is divided into a number of tracks. Total tracks are equal to the size of the paper, i.e. for a paper width of 132 characters, drum will have 132 tracks.

#### **Advantages**

- Very high speed

#### **Disadvantages**

- Very expensive
- Characters fonts cannot be changed

#### **Chain Printer**

In this printer, a chain of character sets is used, hence it is called Chain Printer. A standard character set may have 48, 64, or 96 characters.

#### **Advantages**

- Character fonts can easily be changed.
- Different languages can be used with the same printer.

#### **Disadvantages**

- Noisy

### **Non-impact Printers:-**

Non-impact printers print the characters without using the ribbon. These printers print a complete page at a time, thus they are also called as Page Printers.



These printers are of two types –

- Laser Printers
- Inkjet Printers

### **Characteristics of Non-impact Printers**

- Faster than impact printers
- They are not noisy
- High quality
- Supports many fonts and different character size

### **Laser Printers**

These are non-impact page printers. They use laser lights to produce the dots needed to form the characters to be printed on a page.

### **Advantages**

- Very high speed
- Very high quality output
- Good graphics quality
- Supports many fonts and different character size

### **Disadvantages**

- Expensive
- Cannot be used to produce multiple copies of a document in a single printing

### **Inkjet Printers**

Inkjet printers are non-impact character printers based on a relatively new technology. They print characters by spraying small drops of ink onto paper. Inkjet printers produce high quality output with presentable features.

They make less noise because no hammering is done and these have many styles of printing modes available. Color printing is also possible. Some models of Inkjet printers can produce multiple copies of printing also.

### **Advantages**

- High quality printing
- More reliable

### **Disadvantages**

- Expensive as the cost per page is high
- Slow as compared to laser printer

## **GENERATION OF COMPUTER:-**

A computer is an electronic device that manipulates information or data. It has the ability to store, retrieve, and process data.

Nowadays, a computer can be used to type documents, send email, play games, and browse the Web. It can also be used to edit or create spreadsheets, presentations, and even videos. But the evolution of this complex system started around 1940 with the first Generation of Computer and evolving ever since.

There are five generations of computers.

## 1. FIRST GENERATION

### Introduction:

- 1946-1959 is the period of first generation computer.
- J.P.Eckert and J.W.Mauchy invented the first successful electronic computer called ENIAC, ENIAC stands for “Electronic Numeric Integrated And Calculator”.

### Few Examples are:

- ENIAC
- EDVAC
- UNIVAC
- IBM-701
- IBM-650

### Advantages:

- It made use of vacuum tubes which are the only electronic component available during those days.
- These computers could calculate in milliseconds.

### Disadvantages:

- These were very big in size, weight was about 30 tones.
- These computers were based on vacuum tubes.
- These computers were very costly.
- It could store only a small amount of information due to the presence of magnetic drums.
- As the invention of first generation computers involves vacuum tubes, so another disadvantage of these computers was, vacuum tubes require a large cooling system.
- Very less work efficiency.
- Limited programming capabilities and punch cards were used to take inputs.
- Large amount of energy consumption.
- Not reliable and constant maintenance is required.

## 2. SECOND GENERATION

### Introduction:

- 1959-1965 is the period of second-generation computer.
- 3.Second generation computers were based on Transistor instead of vacuum tubes.

### Few Examples are:

- Honeywell 400
- IBM 7094
- CDC 1604
- CDC 3600
- UNIVAC 1108

### Advantages:

- Due to the presence of transistors instead of vacuum tubes, the size of electron component decreased. This resulted in reducing the size of a computer as compared to first generation computers.
- Less energy and not produce as much heat as the first generation.
- Assembly language and punch cards were used for input.
- Low cost than first generation computers.
- Better speed, calculate data in microseconds.
- Better portability as compared to first generation

### Disadvantages:

- A cooling system was required.
- Constant maintenance was required.
- Only used for specific purposes.

### 3. THIRD GENERATION

#### Introduction:

- 1965-1971 is the period of third generation computer.
- These computers were based on Integrated circuits.
- IC was invented by Robert Noyce and Jack Kilby In 1958-1959.
- IC was a single component containing number of transistors.

#### Few Examples are:

- PDP-8
- PDP-11
- ICL 2900
- IBM 360
- IBM 370

#### Advantages:

- These computers were cheaper as compared to second-generation computers.
- They were fast and reliable.
- Use of IC in the computer provides the small size of the computer.
- IC not only reduce the size of the computer but it also improves the performance of the computer as compared to previous computers.
- This generation of computers has big storage capacity.
- Instead of punch cards, mouse and keyboard are used for input.
- They used an operating system for better resource management and used the concept of time-sharing and multiple programming.
- These computers reduce the computational time from microseconds to nanoseconds.

#### Disadvantages:

- IC chips are difficult to maintain.
- The highly sophisticated technology required for the manufacturing of IC chips.
- Air conditioning is required.

### 4. FOURTH GENERATION

#### Introduction:

- 1971-1980 is the period of fourth generation computer.
- This technology is based on Microprocessor.
- A microprocessor is used in a computer for any logical and arithmetic function to be performed in any program.
- Graphics User Interface (GUI) technology was exploited to offer more comfort to users.

#### Few Examples are:

- IBM 4341
- DEC 10
- STAR 1000
- PUP 11

#### Advantages:

- Fastest in computation and size get reduced as compared to the previous generation of computer.
- Heat generated is negligible.
- Small in size as compared to previous generation computers.
- Less maintenance is required.

- All types of high-level language can be used in this type of computers.

**Disadvantages:**

- The Microprocessor design and fabrication are very complex.
- Air conditioning is required in many cases due to the presence of ICs.
- Advance technology is required to make the ICs.

## 5. FIFTH GENERATION

**Introduction:**

- The period of the fifth generation in 1980-onwards.
- This generation is based on artificial intelligence.
- The aim of the fifth generation is to make a device which could respond to natural language input and are capable of learning and self-organization.
- This generation is based on ULSI(Ultra Large Scale Integration) technology resulting in the production of microprocessor chips having ten million electronic component.

**Few Examples are:**

- Desktop
- Laptop
- Notebook
- Ultra Book
- Chrome book

**Advantages:**

- It is more reliable and works faster.
- It is available in different sizes and unique features.
- It provides computers with more user-friendly interfaces with multimedia features.

**Disadvantages:**

- They need very low-level languages.
- They may make the human brains dull and doomed.

### **CLASSIFICATION OF COMPUTER:-**

The computer systems can be classified on the following basis:

1. On the basis of size.
2. On the basis of mechanism.
3. On the basis of purpose.

#### **1. Classification on the basis of size:-**

It is again classified into following types such as:-

a) **Super computers :-**

b) **Mainframe computer**

c) **Mini computer**

d) **Micro computer**

a) **Super computer:-**

- They are very fast and powerful machines.
- Run at the speed of 10 MIPS (million instructions per second).
- Very expensive.
- Not used for general applications.
- Example : CDC Cyber

b) **Mainframe computer:-**

- They are built for general computing.
- Very fast and will process information at about 10 MIPS.
- Very expensive.
- Not easily found in general design offices.

**c) Mini computer:-**

- They were developed in 1960's resulting from advanced in microchip technology.
- smaller and less expensive than mainframe computers,
- Run at several MIPS and can support 5-20 users.
- Low cost and high performance.

**d) Micro computer:-**

- They were invented in the 1970's
- Used for home computing.
- Dedicated data processing workstations.
- Advanced in technology have improved microcomputer capacities, resulting in the explosive growth of personal computers in industry.

. micro computer includes:-

- Desktop
- Laptop
- Palmtop

Classification on the basis of mechanism:-

It is again classified into three types such as

a) **Analog computer**

b) **Digital computer**

c) **Hybrid computer**

**a) Analog computer:-**

- The word "analog" means continuously changeable in quantity.
- The analog computers accept input data in continuous form and output is obtained in the form of graphs.
- Example: voltage, current, sound, speed, temperature etc
- The analog computers are used to measure the continuous values.
- Example: Thermometer

**b) Digital computer:-**

- The word "digital" means separate. it refers to binary system, which consists of only two digits, i.e 0 and 1.
- Digital data consists of binary data represented by OFF(low) and ON(high) electrical beats.
- In digital computers, quantities are counted rather than measured.
- A digital computer operates by counting numbers or digits and gives output in digital form.

- Example: calculators, personal computers, digital watches etc

**c) Hybrid computer:-**

- They combine the features of both analog and digital computers.
- Analog machines measure patient's temperatures, then they are converted to numbers and supplied to digital computers.
- In hybrid computers, users can process both continuous (analog) and discontinuous (digital) data.
- These are special purpose computers.
- These are very fast and accurate.
- These are used in scientific fields.

**2. Classification on the basis of purpose:-**

It is classified into two types such as

**a) Special purpose computer**

**b) General purpose computer**

**a) Special purpose computer:-**

- These are designed to perform a specific task. Such computers lack flexibility.
- They perform the task for which they are designed very efficiently.
- Example: Air craft control system, missile guidance system etc.

**b) General purpose computer:-**

- A general purpose computer is one that can be used for a variety of applications.
- Its versatility enables execution of programs of almost any type. These are used in business applications.

**Computer memory and its classification:-**

A memory is just like a human brain. It is used to store data and instructions. Computer memory is the storage space in the computer, where data is to be processed and instructions required for processing are stored. The memory is divided into a large number of small parts called cells. Each location or cell has a unique address, which varies from zero to memory size minus one. For example, if the computer has 64k words, then this memory unit has  $64 * 1024 = 65536$  memory locations. The address of these locations varies from 0 to 65535.

Memory is primarily of three types –

- Cache Memory
- Primary Memory/Main Memory
- Secondary Memory

**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.

Advantages

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.

#### Disadvantages

The disadvantages of cache memory are as follows –

- Cache memory has limited capacity.
- It is very expensive.

### **Primary Memory (Main Memory)**

Primary memory holds only those data and instructions on which the computer is currently working. It has a limited capacity and data is lost when power is switched off. It is generally made up of semiconductor device.

**It is divided into two subcategories RAM and ROM.** Characteristics of Main Memory

- These are semiconductor memories.
- It is known as the main memory.
- Usually volatile memory.
- Data is lost in case power is switched off.
- It is the working memory of the computer.
- Faster than secondary memories.
- A computer cannot run without the primary memory.

### **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.

**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. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not be refreshed on a regular basis.

There is extra space in the matrix, hence SRAM uses more chips than DRAM for the same amount of storage space, making the manufacturing costs higher. SRAM is thus used as cache memory and has very fast access.

### **Characteristic of Static RAM**

- Long life
- No need to refresh
- Faster
- Used as cache memory
- Large size
- Expensive
- High power consumption

### **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. DRAM is used for most system memory as it is cheap and small. All DRAMs are made up of memory cells, which are composed of one capacitor and one transistor.

### **Characteristics of Dynamic RAM**

- Short data lifetime
- Needs to be refreshed continuously
- Slower as compared to SRAM
- Used as RAM
- Smaller in size
- Less expensive
- Less power consumption

**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.

Let us now discuss the various types of ROMs and their characteristics.

### **MROM (Masked ROM)**

The very first ROMs were hard-wired devices that contained a pre-programmed set of data or instructions. These kind of ROMs are known as masked ROMs, which are inexpensive.

### **PROM (Programmable Read Only Memory)**



PROM is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM program. Inside the PROM chip, there are small fuses which are burnt open during programming. It can be programmed only once and is not erasable.

### **EPROM (Erasable and Programmable Read Only Memory)**

EPROM can be erased by exposing it to ultra-violet light for a duration of up to 40 minutes. Usually, an EPROM eraser achieves this function. During programming, an electrical charge is trapped in an insulated gate region. The charge is retained for more than 10 years because the charge has no leakage path. For erasing this charge, ultra-violet light is passed through a quartz crystal window (lid). This exposure to ultra-violet light dissipates the charge. During normal use, the quartz lid is sealed with a sticker.

### **EEPROM (Electrically Erasable and Programmable Read Only Memory)**

EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (millisecond). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of reprogramming is flexible but slow.

### **Advantages of ROM**

The advantages of ROM are as follows –

- Non-volatile in nature
- Cannot be accidentally changed
- Cheaper than RAMs
- Easy to test
- More reliable than RAMs
- Static and do not require refreshing
- Contents are always known and can be verified

### **Computer applications:-**

we will discuss the application of computers in various fields.

#### **Business**

A computer has high speed of calculation, diligence, accuracy, reliability, or versatility which has made it an integrated part in all business organizations.

Computer is used in business organizations for –

- Payroll calculations
- Budgeting
- Sales analysis
- Financial forecasting
- Managing employee database
- Maintenance of stocks, etc.

#### **Banking**

Today, banking is almost totally dependent on computers.

Banks provide the following facilities –

- Online accounting facility, which includes checking current balance, making deposits and overdrafts, checking interest charges, shares, and trustee records.
- ATM machines which are completely automated are making it even easier for customers to deal with banks.

### **Insurance**

Insurance companies are keeping all records up-to-date with the help of computers. Insurance companies, finance houses, and stock broking firms are widely using computers for their concerns.

Insurance companies are maintaining a database of all clients with information showing –

- Procedure to continue with policies
- Starting date of the policies
- Next due installment of a policy
- Maturity date
- Interests due
- Survival benefits
- Bonus

### **Education**

The computer helps in providing a lot of facilities in the education system.

- The computer provides a tool in the education system known as CBE (Computer Based Education).
- CBE involves control, delivery, and evaluation of learning.
- Computer education is rapidly increasing the graph of number of computer students.
- There are a number of methods in which educational institutions can use a computer to educate the students.
- It is used to prepare a database about performance of a student and analysis is carried out on this basis.

### **Marketing**

In marketing, uses of the computer are following –

- **Advertising** – With computers, advertising professionals create art and graphics, write and revise copy, and print and disseminate ads with the goal of selling more products.
- **Home Shopping** – Home shopping has been made possible through the use of computerized catalogues that provide access to product information and permit direct entry of orders to be filled by the customers.

### **Healthcare**

Computers have become an important part in hospitals, labs, and dispensaries. They are being used in hospitals to keep the record of patients and medicines. It is also used in scanning and diagnosing different diseases. ECG, EEG, ultrasounds and CT scans, etc. are also done by computerized machines.

Following are some major fields of health care in which computers are used.

- **Diagnostic System** – Computers are used to collect data and identify the cause of illness.
- **Lab-diagnostic System** – All tests can be done and the reports are prepared by computer.
- **Patient Monitoring System** – These are used to check the patient's signs for abnormality such as in Cardiac Arrest, ECG, etc.

- **Pharma Information System** – Computer is used to check drug labels, expiry dates, harmful side effects, etc.
- **Surgery** – Nowadays, computers are also used in performing surgery.

### **Engineering Design**

Computers are widely used for Engineering purpose.

One of the major areas is CAD (Computer Aided Design) that provides creation and modification of images. Some of the fields are –

- **Structural Engineering** – Requires stress and strain analysis for design of ships, buildings, budgets, airplanes, etc.
- **Industrial Engineering** – Computers deal with design, implementation, and improvement of integrated systems of people, materials, and equipment.
- **Architectural Engineering** – Computers help in planning towns, designing buildings, determining a range of buildings on a site using both 2D and 3D drawings.

### **Military**

Computers are largely used in defence. Modern tanks, missiles, weapons, etc. Military also employs computerized control systems. Some military areas where a computer has been used are –

- Missile Control
- Military Communication
- Military Operation and Planning
- Smart Weapons

### **Communication**

Communication is a way to convey a message, an idea, a picture, or speech that is received and understood clearly and correctly by the person for whom it is meant. Some main areas in this category are –

- E-mail
- Chatting
- Usenet
- FTP
- Telnet
- Video-conferencing

### **Government**

Computers play an important role in government services. Some major fields in this category are –

- Budgets
- Sales tax department
- Income tax department
- Computation of male/female ratio
- Computerization of voters lists
- Computerization of PAN card
- Weather forecasting

## Solved Questions

### Short Answer Type Questions.

#### **Outline the key features of 1st Generation Computer in brief. (2019-Winter)**

**Ans.** The period of first generation was from 1946-1959. The computers of first generation used vacuum tubes as the basic components for memory and circuitry for CPU (Central Processing Unit). These tubes, like electric bulbs, produced a lot of heat and the installations used to fuse frequently. Therefore, they were very expensive and only large organizations were able to afford it. In this generation, mainly batch processing operating system was used. Punchcards, paper tape, and magnetic tape was used as input and output devices. The computers in this generation used machine code as the programming language.

#### **What is CPU? (2018-Summer)**

**Ans.** A central processing unit (CPU), also called a central processor, main processor or just processor, is the electronic circuitry within a computer that executes instructions that make up a computer program. The CPU performs basic arithmetic, logic, controlling, and input/output (I/O) operations specified by the instructions in the program.

#### **What is ALU? (2014-Winter)**

**Ans.** Stands for "Arithmetic Logic Unit." An ALU is an integrated circuit within a CPU or GPU that performs arithmetic and logic operations. Arithmetic instructions include addition, subtraction, and shifting operations, while logic instructions include boolean comparisons, such as AND, OR, XOR, and NOT operations.

#### **Name three input devices used in PC. (2016-Summer)**

**Ans.** The three input devices used in PC are Keyboard, Mouse and Scanner.

### Long Answer Type Questions.

#### **Discuss about the generation of computers? Explain the key features of computers of each generation? (2013-Winter)**

**Ans.** A computer is an electronic device that manipulates information or data. It has the ability to store, retrieve, and process data.

Nowadays, a computer can be used to type documents, send email, play games, and browse the Web. It can also be used to edit or create spreadsheets, presentations, and even videos. But the evolution of this complex system started around 1940 with the first Generation of Computer and evolving ever since.

There are five generations of computers.

#### **FIRST GENERATION**

##### **Features:**

- 1946-1959 is the period of first generation computer.
- J.P. Eckert and J.W. Mauchly invented the first successful electronic computer called ENIAC, ENIAC stands for—Electronic Numeric Integrated and Calculator.
- Few Examples are: ENIAC, EDVAC, UNIVAC, etc.

**Advantages:**

- It made use of vacuum tubes which are the only electronic component available during those days.
- These computers could calculate in milliseconds.

**Disadvantages:**

- These were very big in size, weight was about 30 tons.
- These computers were based on vacuum tubes.
- These computers were very costly.

**SECOND GENERATION****Features:**

- 1959-1965 is the period of second-generation computer.
- Second generation computers were based on Transistor instead of vacuum tubes.
- Few examples are: Honeywell 400, IBM 7094, etc.

**Advantages:**

- Due to the presence of transistors instead of vacuum tubes, the size of electron component decreased. This resulted in reducing the size of a computer as compared to first generation computers.
- Less energy and not produce as much heat as the first generation.
- Assembly language and punch cards were used for input.

**Disadvantages:**

- A cooling system was required.
- Constant maintenance was required.
- Only used for specific purposes.

**THIRD GENERATION****Features:**

- 1965-1971 is the period of third generation computer.
- These computers were based on Integrated circuits.
- IC was invented by Robert Noyce and Jack Kilby in 1958-1959.
- IC was a single component containing number of transistors.
- Few examples are: PDP-8, PDP-11, ICL 2900, etc.

**Advantages:**

- These computers were cheaper as compared to second-generation computers.
- They were fast and reliable.
- Use of IC in the computer provides the small size of the computer.

**Disadvantages:**

- IC chips are difficult to maintain.
- The highly sophisticated technology required for the manufacturing of IC chips.
- Air conditioning is required.

**FOURTH GENERATION****Features:**

- 1971-1980 is the period of fourth generation computer.
- This technology is based on Microprocessor.
- A microprocessor is used in a computer for any logical and arithmetic function to be performed in any program.

- GraphicsUserInterface(GUI)technologywasexploitedtooffer more comfort to users.
- Few Examples are: IBM 4341, DEC 10, STAR 1000, etc.

**Advantages:**

- Fastest in computation and size get reduced as compared to the previous generation of computer.
- Heat generated is negligible.
- Small in size as compared to previous generation computers.

**Disadvantages:**

- The Microprocessor design and fabrication are very complex.
- Air conditioning is required in many cases due to the presence of ICs.
- Advance technology is required to make the ICs.

**FIFTH GENERATION**

**Features:**

- The period of the fifth generation is 1980-onwards.
- This generation is based on artificial intelligence.
- The aim of the fifth generation is to make a device which could respond to natural language input and are capable of learning and self-organization.
- This generation is based on ULSI (Ultra Large Scale Integration) technology resulting in the production of microprocessor chips having ten million electronic components.
- Few Examples are: Desktop, Laptop, etc.

**Advantages:**

- It is more reliable and works faster.
- It is available in different sizes and unique features.
- It provides computers with more user-friendly interfaces with multimedia features.

**Disadvantages:**

- They need very low-level languages.
- They may make the human brains dull and doomed.

**Discuss about the various input devices used in PC platform?(2013-Summer) Ans.**

Following are some of the important input devices which are used in a computer –

**Keyboard**

Keyboard is the most common and very popular input device which helps to input data to the computer. The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing additional functions.

Keyboards are of two sizes 84 keys or 101/102 keys, but now keyboards with 104 keys or 108 keys are also available for Windows and Internet.

**Mouse**

Mouse is the most popular pointing device. It is a very famous cursor-control device having a small palm size box with a round ball at its base, which senses the movement of the mouse and sends corresponding signals to the CPU when the mouse buttons are pressed.

Generally, it has two buttons called the left and the right button and a wheel is present between the buttons. A mouse can be used to control the position of the cursor on the screen, but it cannot be used to enter text into the computer.

## **Joystick**

Joystick is also a pointing device, which is used to move the cursor position on a monitor screen. It is a stick having a spherical ball at its both lower and upper ends. The lower spherical ball moves in a socket. The joystick can be moved in all four directions.

## **Joystick**

The function of the joystick is similar to that of a mouse. It is mainly used in Computer Aided Designing (CAD) and playing computer games.

## **Light Pen**

Light pen is a pointing device similar to a pen. It is used to select a displayed menu item or draw pictures on the monitor screen. It consists of a photocell and an optical system placed in a small tube.

When the tip of a light pen is moved over the monitor screen and the pen button is pressed, its photocell senses the element, detects the screen location, and sends the corresponding signal to the CPU.

## **TrackBall**

Trackball is an input device that is mostly used in notebook or laptop computer, instead of a mouse. This is a ball which is half inserted and by moving fingers on the ball, the pointer can be moved.

Since the whole device is not moved, a track ball requires less space than a mouse. A trackball comes in various shapes like a ball, a button, or a square.

## **Scanner**

Scanner is an input device, which works more like a photocopy machine. It is used when some information is available on paper and it is to be transferred to the hard disk of the computer for further manipulation.

Scanner captures images from the source which are then converted into a digital form that can be stored on the disk. These images can be edited before they are printed.

## **Digitizer**

Digitizer is an input device which converts analog information into digital form. Digitizer can convert a signal from the television or camera into a series of numbers that could be stored in a computer. They can be used by the computer to create a picture of whatever the camera had been pointed at.

Digitizer is also known as Tablet or Graphics Tablet as it converts graphics and pictorial data into binary inputs. A graphic tablet as digitizer is used for fine works of drawing and image manipulation applications.

## **Microphone**

Microphone is an input device to input sound that is then stored in a digital form.

The microphone is used for various applications such as adding sound to a multimedia presentation or for mixing music.

## **Magnetic Ink Card Reader (MICR)**

MICR input device is generally used in banks as there are large number of cheques to be processed every day. The bank's code number and cheque number are printed on the cheques with a special type of ink that contains particles of magnetic material that are machine readable.

This reading process is called Magnetic Ink Character Recognition (MICR). The main advantages of MICR is that it is fast and less error prone.

## **OpticalCharacterReader(OCR)**

OCR is an input device used to read a printed text.

OCR scans the text optically, character by character, converts them into a machine-readable code, and stores the text on the system memory.

## **BarCodeReaders**

Bar Code Reader is a device used for reading bar-coded data (data in the form of light and dark lines). Bar-coded data is generally used in labelling goods, numbering the books, etc. It may be a handheld scanner or may be embedded in a stationary scanner.

Bar Code Reader scans a bar code image, converts it into an alphanumeric value, which is then fed to the computer that the bar code reader is connected to.

## **OpticalMarkReader(OMR)**

OMR is a special type of optical scanner used to recognize the type of mark made by pen or pencil. It is used where one out of a few alternatives is to be selected and marked.

It is specially used for checking the answer sheets of examinations having multiple-choice questions.

## **EXERCISE**

### **Short Answer Type Questions.**

Outline the key features of 4th generation computer in brief.

Differentiate between analog computer and digital computer.

Define hybrid computer.

Differentiate between data and information?

Differentiate between soft copy and hard copy output?

Define Auxiliary memory used in PC.

Define primary memory used in PC.

Define memory hierarchy. State the different memory present in the different levels.

*(2015-Winter, 2013-Winter)*

Define cache memory.

What is an OMR?

What is a MICR? *(2014-Winter)*

Define Hard copy output and Soft copy output from a PC.



Differentiate between volatile and non-volatile computer memory.

What are the different types of plotters used in PC?

### **Long Answer Type Questions.**

Discuss about the various output devices used in a PC platform?

What is classification of computer? Compare and contrast the features of different classes of computers? *(2013-Winter)*

Discuss about the evolution of computers? Give suitable examples of various computers?

What is memory hierarchy? Explain the main features of the various types of memory present at different levels of this hierarchy? *(2016-Winter)*

Differentiate between primary memory and secondary memory used in PC. Give proper examples to substantiate your answer? *(2017-Winter)*

\*\*\*\*\*

# CHAPTER-2:COMPUTERSOFTWARE

Software is a set of programs, which is designed to perform a well-defined function. A program is a sequence of instructions written to solve a particular problem.

There are two types of software –

- System Software
- Application Software

## System Software

The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself. System software is generally prepared by the computer manufacturers. These software products comprise of programs written in low-level languages, which interact with the hardware at a very basic level. System software serves as the interface between the hardware and the end users.

Some examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc.

Here is a list of some of the most prominent features of a system software –

- Close to the system
- Fast in speed
- Difficult to design
- Difficult to understand
- Less interactive
- Smaller in size
- Difficult to manipulate
- Generally written in low-level language

## Application Software

Application software products are designed to satisfy a particular need of a particular environment. All software applications prepared in the computer lab can come under the category of Application software.

Application software may consist of a single program, such as Microsoft's notepad for writing and editing a simple text. It may also consist of a collection of programs, often called a software package, which work together to accomplish a task, such as a spreadsheet package.

Examples of Application software are the following –

- Payroll Software
- Student Record Software
- Inventory Management Software
- Income Tax Software
- Railways Reservation Software
- Microsoft Office Suite Software

- Microsoft Word
- Microsoft Excel
- Microsoft PowerPoint

Features of application software are as follows –

- Close to the user
- Easy to design
- More interactive
- Slow in speed
- Generally written in high-level language
- Easy to understand
- Easy to manipulate and use
- Bigger in size and requires large storage space

### **What is an Operating System?**

An Operating System (OS) is a software that acts as an interface between computer hardware components and the user. Every computer system must have at least one operating system to run other programs. Applications like Browsers, MS Office, Notepad Games, etc., need some environment to run and perform its tasks.

The OS helps you to communicate with the computer without knowing how to speak the computer's language. It is not possible for the user to use any computer or mobile device without having an operating system.

### **History Of OS:-**

- Operating systems were first developed in the late 1950s to manage tape storage
- The General Motors Research Lab implemented the first OS in the early 1950s for their IBM 701
- In the mid-1960s, operating systems started to use disks
- In the late 1960s, the first version of the Unix OS was developed
- The first OS built by Microsoft was DOS. It was built in 1981 by purchasing the 86-DOS software from a Seattle company
- The present-day popular OS Windows first came to existence in 1985 when a GUI was created and paired with MS-DOS.

OS Name	Share
Windows	40.34
Android	37.95
iOS	15.44
Mac OS	4.34
Linux	0.95

Chrome OS 0.14

Windows Phone OS 0.06

### **Types of Operating System (OS):-**

Following are the popular types of Operating System:

- Batch Operating System
- Multitasking/Time Sharing OS
- Multiprocessing OS
- Real Time OS
- Distributed OS
- Network OS
- Mobile OS

#### **Batch Operating System**

Some computer processes are very lengthy and time-consuming. To speed the same process, a job with a similar type of needs are batched together and run as a group.

The user of a batch operating system never directly interacts with the computer. In this type of OS, every user prepares his or her job on an offline device like a punch card and submit it to the computer operator.

#### **Multi-Tasking/Time-sharing Operating systems**

Time-sharing operating system enables people located at a different terminal(shell) to use a single computer system at the same time. The processor time (CPU) which is shared among multiple users is termed as time sharing.

#### **Real time OS**

A real time operating system time interval to process and respond to inputs is very small. Examples: Military Software Systems, Space Software Systems are the Real time OS example.

#### **Distributed Operating System**

Distributed systems use many processors located in different machines to provide very fast computation to its users.

#### **Network Operating System**

Network Operating System runs on a server. It provides the capability to serve to manage data, user, groups, security, application, and other networking functions.

#### **Mobile OS**

Mobile operating systems are those OS which is especially that are designed to power smartphones, tablets, and wearables devices.

Some most famous mobile operating systems are Android and iOS, but others include BlackBerry, Web, and watchOS.

### **Functions of Operating System**

In an operating system software performs each of the function:

1. **Process management:-** Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.
2. **Memory management:-** Memory management module performs the task of allocation and de-allocation of memory space to programs in need of this resources.
3. **File management:-** It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.

4. **Device Management:** Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and de-allocation of the devices.
5. **I/O System Management:** One of the main objects of any OS is to hide the peculiarities of that hardware devices from the user.
6. **Secondary-Storage Management:** Systems have several levels of storage which includes primary storage, secondary storage, and cache storage. Instructions and data must be stored in primary storage or cache so that a running program can reference it.
7. **Security:-** Security module protects the data and information of a computer system against malware threat and authorized access.
8. **Command interpretation:** This module is interpreting commands given by the and acting system resources to process that commands.
9. **Networking:** A distributed system is a group of processors which do not share memory, hardware devices, or a clock. The processors communicate with one another through the network.
10. **Job accounting:** Keeping track of time & resource used by various job and users.
11. **Communication management:** Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

### **Features of Operating System (OS)**

Here is a list important features of OS:

- Protected and supervisor mode
- Allows disk access and file systems Device drivers Networking Security
- Program Execution
- Memory management Virtual Memory Multitasking
- Handling I/O operations
- Manipulation of the file system
- Error Detection and handling
- Resource allocation
- Information and Resource Protection

### **Advantage of using Operating System**

- Allows you to hide details of hardware by creating an abstraction
- Easy to use with a GUI
- Offers an environment in which a user may execute programs/applications
- The operating system must make sure that the computer system convenient to use
- Operating System acts as an intermediary among applications and the hardware components
- It provides the computer system resources with easy to use format
- Acts as an intermediary between all hardware's and software's of the system

### **Disadvantages of using Operating System**

- If any issue occurs in OS, you may lose all the contents which have been stored in your system
- Operating system's software is quite expensive for small size organization which adds burden on them. Example Windows
- It is never entirely secure as a threat can occur at any time

### **What is a Kernel?**

The kernel is the central component of a computer operating systems. The only job performed by the kernel is to the manage the communication between the software and the hardware. A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.

### Introduction to Kernel

#### Features of Kennel

- Low-level scheduling of processes
- Inter-process communication
- Process synchronization
- Context switching

#### Types of Kernels

There are many types of kernels that exists, but among them, the two most popular kernels are:

##### 1. Monolithic

A monolithic kernel is a single code or block of the program. It provides all the required services offered by the operating system. It is a simplistic design which creates a distinct communication layer between the hardware and software.

##### 2. Microkernels

Microkernel manages all system resources. In this type of kernel, services are implemented in different address space. The user services are stored in user address space, and kernel services are stored under kernel address space. So, it helps to reduce the size of both the kernel and operating system.

#### Difference between Firmware and Operating System

Firmware	Operating System
Firmware is one kind of programming that is embedded on a chip in the device which controls that specific device.	OS provides functionality over and above that which is provided by the firmware.
Firmware is programs that been encoded by the manufacture of the IC or something and cannot be changed.	OS is a program that can be installed by the user and can be changed.
It is stored on non-volatile memory.	OS is stored on the hard drive.

#### Difference between 32-Bit vs. 64 Bit Operating System

Parameters	32. Bit	64. Bit
Architecture and Software	Allow 32 bit of data processing simultaneously	Allow 64 bit of data processing simultaneously
Compatibility	32-bit applications require 32-bit OS and CPUs.	64-bit applications require a 64-bit OS and CPU.
Systems	All versions of Windows 8,	Windows XP

Available	Windows 7, Windows Vista, and Windows XP, Linux, etc.	Professional, Vista, 7, Mac OS X and Linux.
Memory Limits	32-bit systems are limited to 3.2 GB of RAM.	64-bit systems allow a maximum 17 Billion GB of RAM.

### Disk Operating System

MS-DOS is one of the oldest and widely used operating system. DOS is a set of computer programs, the major functions of which are file management, allocation of system resources, providing essential features to control hardware devices. DOS commands can be typed in either upper case or lower case.

#### Features of DOS

Following are the significant features of DOS –

- It is a single user system.
- It controls program.
- It is machine independence.
- It manages (computer) files.
- It manages input and output system.
- It manages (computer) memory.
- It provides command processing facilities.
- It operates with Assembler.

#### Types of DOS Commands

Following are the major types of DOS Command –

- Internal Commands – Commands such as DEL, COPY, TYPE, etc. are the internal commands that remain stored in computer memory.
- External Commands – Commands like FORMAT, DISKCOPY, etc. are the external commands and remain stored on the disk.

### Windows Operating System

The operating system window is the extension of the disk operating system.

It is the most popular and simplest operating system; it can be used by any person who can read and understand basic English, as it does not require any special training.

However, the Windows Operating System requires DOS to run the various application programs initially. Because of this reason, DOS should be installed into the memory and then window can be executed.

#### Elements of Windows OS

Following are the significant element of Windows Operating System (WOS) –

- Graphical User Interface
- Icons (pictures, documents, application, program icons, etc.)
- Taskbar
- Start button
- Windows explorer
- Mouse button
- Hardware compatibility
- Software compatibility
- Help, etc.

#### Versions of Windows Operating System

Following are the different versions of Windows Operating System –

Version	Year	Version	Year
Window 1.01	1985	Windows XP Professional x64	2005
Windows NT 3.1	1993	Windows Vista	2007
Windows 95	1995	Windows 7	2009
Windows 98	1998	Windows 8	2012
Windows 2000	2000	Windows 10	2015
Windows ME	2000	Windows Server 2016	2016
Windows XP	2001		

### Unix Operating System

The Unix Operating System is the earliest operating system developed in 1970s. Let us consider the following points relating to the Unix Operating System –

- It is an operating system that has multitasking features.
- It has multiuser computer operating systems.
- It runs practically on every sort of hardware and provides stimulus to the open source movement.
- It has comparative complex functionality and hence an untrained user cannot use it; only the one who has taken training can use this system.
- Another drawback of this system is, it does not give notice or warn about the consequences of a user's action (whether user's action is right or wrong).

#### **What is Compiler?**

A compiler is a computer program that transforms code written in a high-level programming language into the machine code. It is a program which translates the human-readable code to a language a computer processor understands (binary 1 and 0 bits). The computer processes the machine code to perform the corresponding tasks.

A compiler should comply with the syntax rule of that programming language in which it is written. However, the compiler is only a program and cannot fix errors found in that program. So, if you make a mistake, you need to make changes in the syntax of your program. Otherwise, it will not compile.

#### **What is Interpreter?**

An interpreter is a computer program, which converts each high-level program statement into the machine code. This includes source code, pre-compiled code, and scripts. Both compiler and interpreters do the same job which is converting higher level programming language to machine code. However, a compiler will convert the code into machine code (create an exe) before program run. Interpreters convert code into machine code when the program is run.

#### Difference Between Compiler and Interpreter

Basis of difference	Compiler	Interpreter
Programming	• Create the program.	• Create the Program



Basis of difference	Compiler	Interpreter
Steps	<ul style="list-style-type: none"> <li>• Compile will parse or analyses all of the language statements for its correctness. If incorrect, throws an error</li> <li>• If no error, the compiler will convert source code to machine code.</li> <li>• It links different code files into a runnable program(know as exe)</li> <li>• Run the Program</li> </ul>	<ul style="list-style-type: none"> <li>• No linking of files or machine code generation</li> <li>• Source statements executed line by line DURING Execution</li> </ul>
Advantage	The program code is already translated into machine code. Thus, it code execution time is less.	Interpreters are easier to use, especially for beginners.
Disadvantage	You can't change the program without going back to the source code.	Interpreted programs can run on computers that have the corresponding interpreter.
Machine code	Store machine language as machine code on the disk	Not saving machine code at all.
Running time	Compiled code run faster	Interpreted code run slower
Model	It is based on language translationlinking-loading model.	It is based on Interpretation Method.
Program generation	Generates output program (in the form of exe) which can be run independently from the original program.	Do not generate output program. So they evaluate the source program at every time during execution.
Execution	Program execution is separate from the compilation. It performed only after the entire output program is compiled.	Program Execution is a part of Interpretation process, so it is performed line by line.
Memory requirement	Target program executeindependently and do not require the compiler in the memory.	The interpreter exists in the memory during interpretation.
Best suited for	Bounded to the specific target machine and cannot be ported. C and C++ are a most popular a programming language which uses compilation	For web environments, where load times are important. Due to all the exhaustive analysis is done, compiles take relatively larger time to compile even small code that may not be run multiple

Basis of difference	Compiler	Interpreter
	model.	times. In such cases, interpreters are better.
Code Optimization	The compiler sees the entire code upfront. Hence, they perform lots of optimizations that make code run faster	Interpreters see code line by line, and thus optimizations are not as robust as compilers
Dynamic Typing	Difficult to implement as compilers cannot predict what happens at run time.	Interpreted languages support Dynamic Typing
Usage	It is best suited for the Production Environment	It is best suited for the program and development environment.
Error execution	Compiler displays all errors and warning at the compilation time. Therefore, you can't run the program without fixing errors	The interpreter reads a single statement and shows the error if any. You must correct the error to interpret next line.
Input	It takes an entire program	It takes a single line of code.
Output	Compilers generate intermediate machine code.	Interpreter never generate any intermediate machine code.
Errors	Display all errors after, compilation, all at the same time.	Displays all errors of each line one by one.
Pertaining Programming languages	C, C++, C#, Scala, Java all use compiler.	PHP, Perl, Ruby use an interpreter.

### Computer virus

A computer virus, much like a flu virus, is designed to spread from host to host and has the ability to replicate itself. Similarly, in the same way that flu viruses cannot reproduce without a host cell, computer viruses cannot reproduce and spread without programming such as a file or document. What are the different types of computer viruses?

#### 1. **Boot sector virus**

This type of virus can take control when you start — or boot — your computer. One way it can spread is by plugging an infected USB drive into your computer.

#### 2. **Web scripting virus**

This type of virus exploits the code of web browsers and web pages. If you access such a web page, the virus can infect your computer.

#### 3. **Browser hijacker**

This type of virus “hijacks” certain web browser functions, and you may be automatically directed to an unintended website.

#### **4. Resident virus**

This is a general term for any virus that inserts itself in a computer system's memory. A resident virus can execute anytime when an operating system loads.

#### **5. Direct action virus**

This type of virus comes into action when you execute a file containing a virus. Otherwise, it remains dormant.

#### **6. Polymorphic virus**

A polymorphic virus changes its code each time an infected file is executed. It does this to evade antivirus programs.

#### **7. File infector virus**

This common virus inserts malicious code into executable files — files used to perform certain functions or operations on a system.

#### **8. Multipartite virus**

This kind of virus infects and spreads in multiple ways. It can infect both program files and system sectors.

#### **9. Macro virus**

Macro viruses are written in the same macro language used for software applications. Such viruses spread when you open an infected document, often through email attachments.

### **How Does Virus Affect?**

Let us discuss in what ways a virus can affect your computer system. The ways are mentioned below –

- By downloading files from the Internet.
- During the removable of media or drives.
- Through pen drive.
- Through e-mail attachments.
- Through unpatched software & services.
- Through unprotected or poor administrator passwords.

### **Impact of Virus**

Let us now see the impact of virus on your computer system –

- Disrupts the normal functionality of respective computer system.
- Disrupts system network use.
- Modifies configuration setting of the system.
- Destroys data.
- Disrupts computer network resources.
- Destroys of confidential data.

### **Virus Detection**

The most fundamental method of detection of virus is to check the functionality of your computer system; a virus affected computer does not take command properly.

However, if there is antivirus software in your computer system, then it can easily check programs and files on a system for virus signatures.

### **Virus Preventive Measures**

Let us now see the different virus preventive measures. A computer system can be protected from virus through the following –

- Installation of an effective antivirus software.
- Patching up the operating system.
- Patching up the client software.

- Putting highly secured Passwords.
- Use of Firewalls.

### **Most Effective Antivirus**

Following are the most popular and effective antivirus from which you can choose one for your personal computer –

- McAfee Antivirus Plus
- Symantec Norton Antivirus
- Avast Pro Antivirus
- Bitdefender Antivirus Plus
- Kaspersky Anti-Virus
- Avira Antivirus
- Webroot Secure Anywhere Antivirus
- Emsisoft Anti-Malware
- Quick Heal Antivirus
- ESET NOD32 Antivirus

### **What are the signs of a computer virus?**

A computer virus attack can produce a variety of symptoms. Here are some of them:

- Frequent pop-up windows. Pop-ups might encourage you to visit unusual sites. Or they might prod you to download antivirus or other software programs.
- Changes to your homepage. Your usual homepage may change to another website, for instance. Plus, you may be unable to reset it.
- Mass emails being sent from your email account. A criminal may take control of your account or send emails in your name from another infected computer.
- Frequent crashes. A virus can inflict major damage on your hard drive. This may cause your device to freeze or crash. It may also prevent your device from coming back on.
- Unusually slow computer performance. A sudden change of processing speed could signal that your computer has a virus.
- Unknown programs that start up when you turn on your computer. You may become aware of the unfamiliar program when you start your computer. Or you might notice it by checking your computer's list of active applications.
- Unusual activities like password changes. This could prevent you from logging into your computer.

### **Solved Questions**

#### **Short Answer Type Questions.**

**Q1 Define an operating system. Give examples of three operating systems used on PC.**

**Ans.** An Operating System (OS) is an interface between computer user and computer hardware. An operating system is 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.

Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

**Q2 Define a batch processing system.**

**Ans.** Batch processing is the processing of transactions in a group or batch. No user interaction is required once batch processing is underway. This differentiates batch processing from transaction processing, which involves processing transactions one at a time and requires user interaction.

**Q3 Define multiprogramming operating system.**

**Ans.** Multiprogramming is a rudimentary form of parallel processing in which several programs are run at the same time on a uniprocessor. Instead, the operating system executes part of one program, then part of another, and so on. To the user it appears that all programs are executing at the same time.

**Q4 Define time sharing operating system.**

**Ans.** Time-

sharing enables many people, located at various terminals, to use a particular computer system at the same time. Multitasking or Time-Sharing Systems is a logical extension of multiprogramming. Processor's time is shared among multiple users simultaneously, termed as time-sharing.

**Long Answer Type Questions****Q1 Define an operating system. Discuss about the major function of any operating system. (2013-Winter)(2017-Winter)**

**Ans.**

An operating system is a program on which application programs are executed and acts as a communication bridge (interface) between the user and the computer hardware.

The main task an operating system carries out is the allocation of resources and services, such as allocation of: memory, devices, processors and information. The operating system also includes programs to manage these resources, such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

**Functions of an operating system:****Security–**

The operating system uses password protection to protect user data and similar other techniques. It also prevents unauthorized access to programs and user data.

**Control over system performance–**

Monitors overall system health to help improve performance records the response time between service requests and system response to have a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.

**Job accounting–**

Operating system keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of users.

**Error detecting aids–**

Operating system constantly monitors the system to detect errors and avoid the malfunctioning of computer system.

**Coordination between other software and users–**

Operating systems also coordinate and assign interpreters, compilers, assemblers and other software to the various users of the computer systems.

## **Memory Management–**

The operating system manages the Primary Memory or Main Memory. Main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address. Main memory is a fast storage and it can be accessed directly by the CPU. For a program to be executed, it should be first loaded in the main memory. An Operating System performs the following activities for memory management:

It keeps tracks of primary memory, i.e., which bytes of memory are used by which user program. The memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used. In multi programming, the OS decides the order in which process are granted access to memory, and for how long. It allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing an I/O operation.

## **Processor Management–**

In a multi programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has. This function of OS is called process scheduling. An Operating System performs the following activities for processor management.

Keeps tracks of the status of processes. The program which perform this task is known as traffic controller. Allocates the CPU that is processor to a process. De-allocates processor when a process is no more required.

## **Device Management–**

An OS manages device communication via their respective drivers. It performs the following activities for device management. Keeps tracks of all devices connected to system. Designates a program responsible for every device known as the Input/output controller. Decides which process gets access to a certain device and for how long. Allocates devices in an effective and efficient way. Deallocates devices when they are no longer required.

## **File Management –**

A file system is organized into directories for efficient or easy navigation and usage. These directories may contain other directories and other files. An Operating System carries out the following file management activities. It keeps track of where information is stored, user access settings and status of every file and more. These facilities are collectively known as the file system.

**Q2** What is the difference between Application Software and System Software? (2015-Winter)

**Ans**

Sr. No.	Key	System Software.	Application Software.
1	Definition	System Software is the type of software which is the interface between application software and system.	On other hand Application Software is the type of software which runs as per user request. It runs on the platform which is provided by system software.
2	Development Language	In general System software are developed in low level language which is more compatible with the system hardware in order to interact with.	While in case of Applications of software high level language is used for their development as they are developed as some specific purpose software.
3	Usage	System software is used for operating computer hardware.	On other hand Application software is used by user to perform specific task.
4	Installation	System software are installed on the computer when operating system is installed.	On other hand Application software are installed according to user's requirements.
5	User interaction	As mentioned in above points system software are specific to system hardware so less or no user interaction available in case of system software.	On other hand in applications of software user can interact with it as user interface is available in this case.
6	Dependency	System software can run independently. It provides platform for running application Software.	On other hand in applications of software can't run independently. They can't run without the presence of system software
7	Examples	Some examples of system software's are compiler, assembler, debugger, driver, etc.	On other hand some examples of application software's are word processor, web browser, media player, etc.

## **EXERCISE**

### **Short Answer Type Questions.**

- Q1 What are the four major functions of an operating system?
- Q2 What are the various types of operating systems used on PC?
- Q3 Define a multi-tasking operating system.
- Q4 Define GUI.
- Q5 What is computer security?
- Q6** What is software? **(2015-Summer)**
- Q7 Write types of software?
- Q8 What is application software?
- Q9 What is system software? Quote some examples of system software?
- Q10 What is an interpreter? How is it different from a compiler?

### **Long Answer Type Questions**

- Q1 Discuss the various objectives of an operating system. Illustrate your answer with proper examples.
- Q2 What is DOS? Discuss about the main features of DOS with appropriate examples.
- Q3 Compare and contrast between the features of UNIX operating system.
- Q4** Compare the various features of Windows and UNIX operating system. Give suitable examples to substantiate your answer. **(2015-Summer)**
- Q5 Write a short note on Single user Vs. Multi-user O.S.
- Q6** What is Virus? How does virus spread and what are the symptoms of virus attack? How can you prevent virus attack? **(2017-Winter)(2013-Winter)(2014-Winter)**
- Q7** Distinguish between Compiler and Interpreter? **(2016-Summer)**
- Q8 Define Software. Describe various types of software and explain them?

\*\*\*\*\*



# CHAPTER-3:COMPUTERNETWORKANDINTERNET

## 3.1 Networkingconcept,Protocol,

### ConnectingMedia,DataTransmissionmode

#### Networkingconcept

- A computer network is a collection of two or more computers, which are connected to two or more computers, which are connected together to share information and resources.
- It is a combination of hardware and software that allows communication between computers over a network.

#### 3.1.1 Protocol

- A protocol is a set of rules that the communication set of rules that the communication between computers on a network.
- Most important set of internet protocols are TCP/IP, HTTPS, SMTP, and FTP.

#### 3.1.2 ConnectingMedia

- Connecting media of a network refer to the transmission media used in the network.
- It refers to the physical media through which communication signals can be transmitted from one point to another.
- It can be divided into two broad categories
  - 1) Guided Media
  - 2) Unguided Media

##### GuidedMedia:

- The data signaling guided medium is bound by the cabling system that guides the data signal along a specific path.
- It consists of a cable composed of materials like copper, tin or silver.
- Basically, they are divided into three categories:

##### (a) Ethernet cable or Twisted Pair:

- In this pair, wires are twisted together, which are surrounded by an insulating material and an outer layer called jacket.
- A twisted pair consists of two conductors (copper)
- Ex- LAN Cable

##### (b) Coaxial Cable:

- It carries the signal of higher frequency data communication through the network.
- It is commonly used in transferring multi-channel television signals in cities.
- Ex-cable TV network

##### (c) Fiber- Optics Cable:

- It is made up of glass or plastic and transmits signals in the form of light from a source to a receiver.
- The speed of optical fiber is hundreds of times faster than coaxial cables.
- Ex-Wavelength Division Multiplexing, SONET network

## Unguided Media:

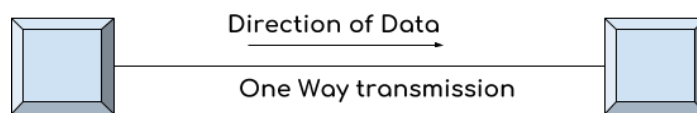
- It is the transfer of information over a distance without the use of enhanced electrical conductors or wires.
- When the computers in a network are interconnected and data is transmitted through waves, then they are said to be connected through unguided media.
- Some commonly used unguided media of transmission are:
  1. Radiowave transmission
  2. Micro wave transmission
  3. Satellite communication
  4. Infrared wave transmission
  5. Bluetooth.

### 3.1.3 Data Transmission mode

- The way in which data is transmitted from one place to another is called Data transmission mode or data communication mode or directional modes.
- There are mainly 3 types of data transmission modes are:
  1. Simplex mode
  2. Half-duplex Mode
  3. Full-duplex mode

#### Simplex mode-

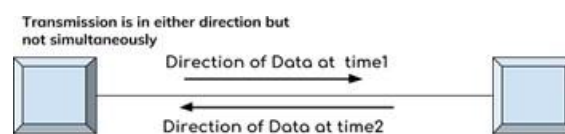
- In simplex mode, data can flow in only one direction.
- In this mode, a sender can only send data and cannot receive it.
- Similarly, a receiver can only receive data but cannot send it.
- Ex-Radio, Television etc.



Simplex Mode

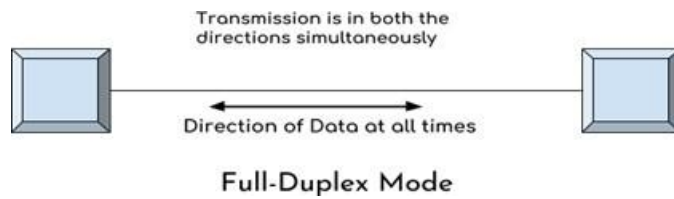
#### Half-duplex Mode-

- In half-duplex mode, data can flow in both directions but not at the same time.
- In this mode, data is sent and received alternatively.
- Ex-walkie-Talkie



Half-Duplex Mode

## Full-Duplex Mode



- In full Duplex-mode, data can flow in both directions at the same time.
- Ex-Mobilephone.

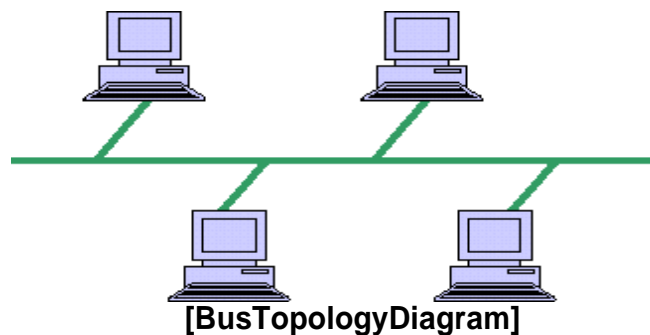
## 3.2 Network Topologies, Types of Network

### 3.2.1 Network Topologies

- Network topology is determined only by the configuration of connections between nodes.
- In a fully connected network with  $n$  nodes, there are  $n(n-1)/2$  direct links.
- The most commonly used topologies are described below:

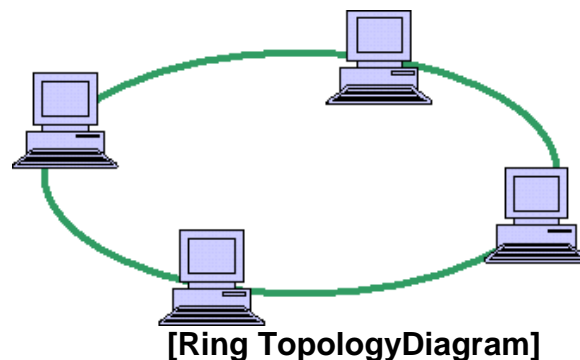
#### 1. Bus Topology

- In case of Bus topology, all devices share a single communication line or cable.
- It is one of the simple forms of networking where a failure of a device does not affect the other devices.
- But failure of the shared communication line can make all other devices stop functioning.



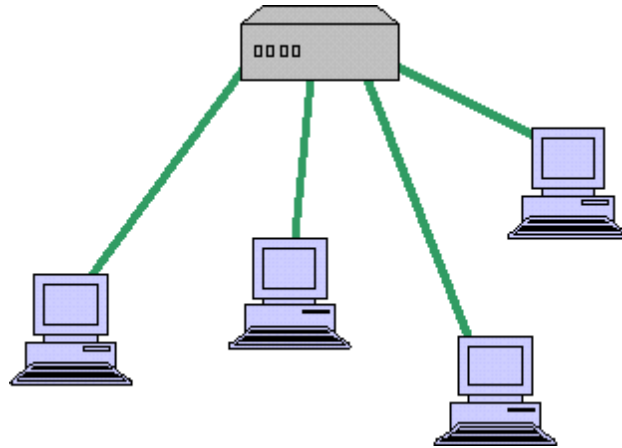
#### 2. Ring Topology

- In ring topology, each host machine connects to exactly two other machines, creating a circular network structure.
- Failure of any host results in failure of the whole ring.
- Thus, every connection in the ring is a point of failure.



### 3. StarTopology

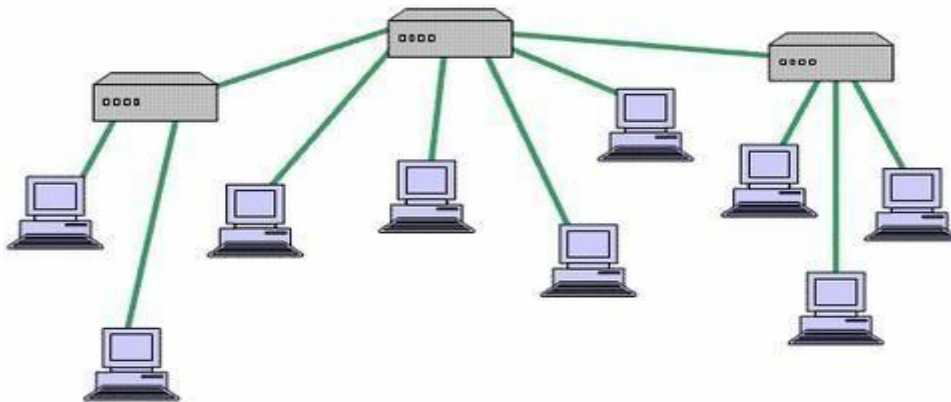
- All hosts in star topology are connected to a central device, known as hub device, using a point-to-point connection.
- If the central hub fails, then the whole network fails.



[Star Topology Diagram]

### 4. TreeTopology

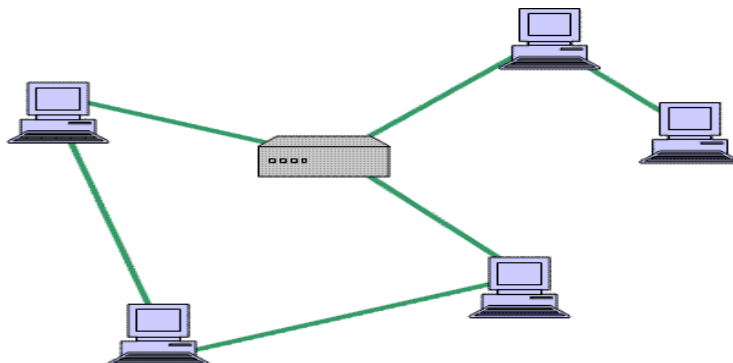
- This is also known as hierarchical topology.
- It is a network topology in which nodes are arranged as a tree.
- The function of the central node in this topology may be distributed.
- It is an extension and variation of star topology.
- A tree topology combines characteristics of linear bus and star topologies.



[Tree Topology Diagram]

### 5. MeshTopology

- It is also known as a completely interconnected topology.
- In mesh topology, every node has a dedicated point-to-point link to every other node.



[Mesh Topology Diagram]

### 3.2.2 Types of Network

- There are broadly classified into three types of computer network.
  - A. LAN
  - B. WAN
  - C. MAN
  - D. PAN

#### LAN (Local Area Network)

- LAN is a small and single site network.
- A LAN connects network devices over a relatively short distance.
- It is a system in which computers are interconnected and the geographical areas such as home, office, building, school may be within a building.
- On most LANs cables are used to connect the computers.
- Data transfer rate in LAN is of the order 10 to 100 Mbps.

#### WAN (Wide Area Network)

- A WAN is a geographically dispersed collection of LANs
- A WAN like the internet spans most of the world.
- A network device called a router connects LAN to a WAN.
- These kind of networks use telephone lines, satellite links and other long-range communication technologies to connect.
- WAN uses technology like ATM for connectivity.

#### MAN (Metropolitan Area Network)

- It is a data network designed for a town or city.
- It connects an area larger than a LAN, but smaller than a WAN, such as a city, with high performance hardware.
- Its main purpose is to share hardware and software resources by the various users.
- Ex- Cable TV network.
- The computers in a WAN are connected using coaxial cables or fiber optic cables.

#### PAN (Personal Area Network)

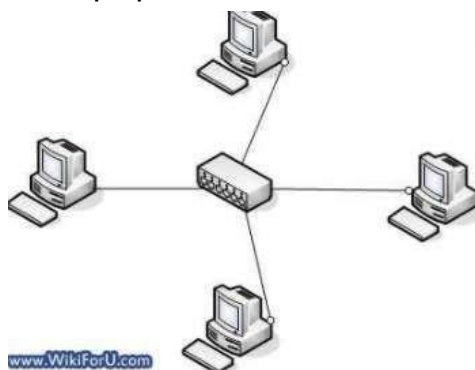
- A personal area network (PAN) is a computer network used for communication among computer devices close to one person.
- Some examples of devices that are used in a PAN are printers, fax machines, telephones, PDAs or scanners.
- The reach of a PAN is typically within about 20-30 feet (approximately 6-9 meters).
- Personal area networks may be wired with computer buses such as USB and FireWire.
- A wireless personal area network (WPAN) can also be made possible with network technologies such as IrDA and Bluetooth..

### 3.3 Networking Devices like Hub, Repeater, Switch, Bridge, Router, Gateway & NIC

- Network devices are required to provide an interface to connect multiple computers in a network.
- There are many types of network devices used in networking. These are Hub, Repeater, Switch, Bridge, Router, Gateway & NIC

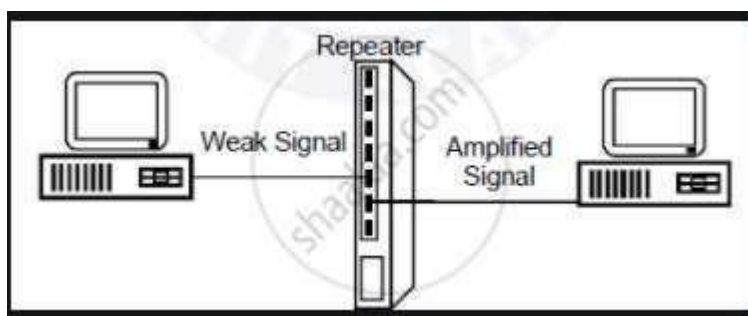
#### HUB-

- Networking using a star topology requires a central point for the devices to connect.
- It is like a repeater with multiple ports to connect the network channels.



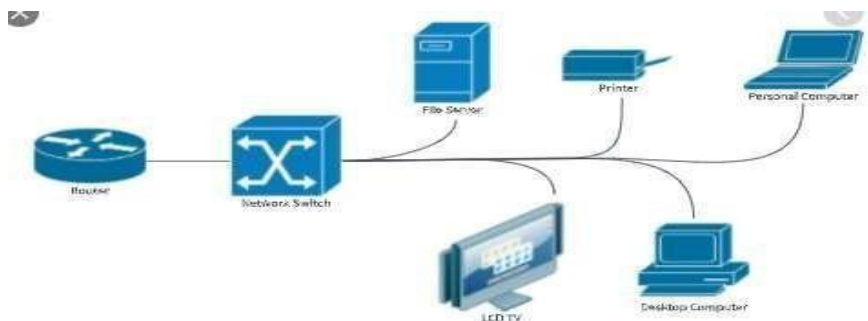
#### REPEATER

- Repeaters have two ports and can connect two segments of a LAN.
- It is an electronic device that receives a signal and retransmits it.



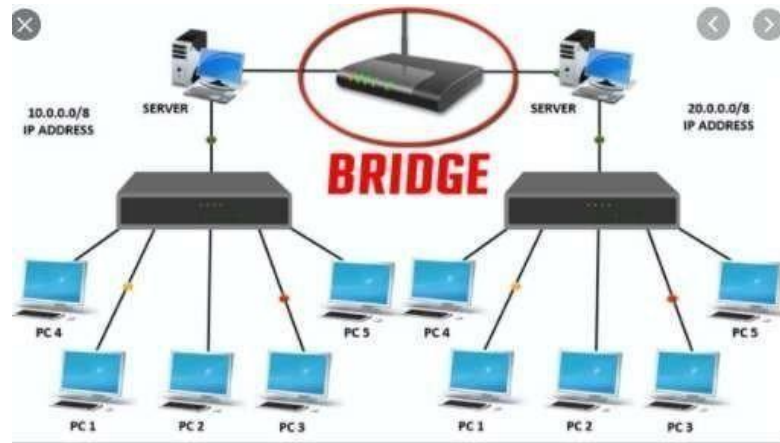
#### SWITCH

- It is a small hardware device that joins multiple computers together on a LAN.
- It helps to reduce overall network traffic.



## BRIDGE

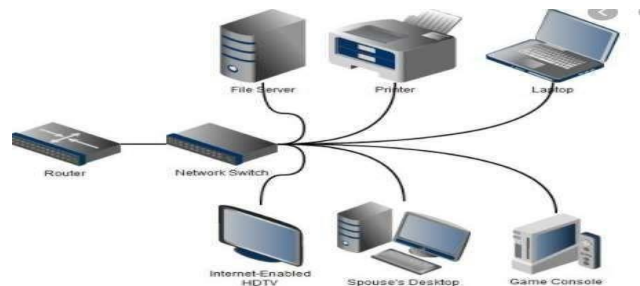
- A bridge is used to join two network segments together; it allows computers on either segment to access resources on the other.
- They can also be used to divide large networks into smaller segments.



- A bridge is used to join two network segments together; it allows computers on either segment to access resources on the other.
- They can also be used to divide large networks into smaller segments.

## ROUTER

- Routers are networking devices used to extend or segment packets from one logical network to another.
- Routers are most often used in large internet networks that use LANs to the internet using dedicated leased lines.

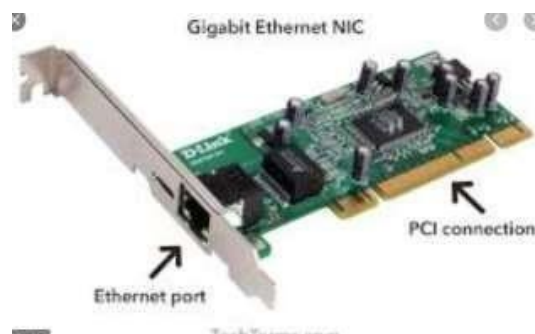


## GATEWAY

- It is an internet networking device, which joins two different network protocols together.
- They are also known as protocol converters.

## NIC (Network Interface Card)

- It is a hardware card installed in a computer so it can communicate on a network.
- The network adapter provides one or more ports for the network cable to connect to and it transmits and receives data onto the network cable.



### 3.4 Internet Services like E-Mail, WWW, FTP, Chatting, Internet

#### Conferencing, Electronic Newspaper & Online Shopping

- Internet is a network of networks that consists of millions of private and public networks of local to global scope.
- An internet user can access a wide variety of services such as

#### E-MAIL (Electronic Mail)

- E-mail is an electronic version of sending and receiving a letter.
  - To use E-mail, a user must have an email address.
  - The email address contains all information required to send or receive a message; it is called a mailbox.
  - Email addresses consist of two parts separated by an @ symbol
    - 1st part is username
    - 2nd part is host name
- (domain name) Example:  
principalbose@rediffmail.com

#### WWW (World Wide Web)

- The World Wide Web is a system of internet servers that supports hypertext and multimedia to access several internet protocols on a single interface.
- It is a way of exchanging information between computers on the internet.

#### Example:

<https://www.google.com/http://www.bosecuttack.in/>

#### FTP (File Transfer Protocol)

- FTP is the internet file transfer between any computer that has an internet connection and also works between computers using totally different operating systems.



## Chatting

- It is a protocol through which internet users can upload files from their computer to a website or download files from a website to their PC.
- It is the easiest way to transfer files between computers via the internet and utilities TCP/IP systems to perform uploading and downloading tasks.
- Chatting is the online textual or multimedia conversation.
- Chatting i.e., a virtual means of communication that involves the sending and receiving of messages, share audio and video between users located in any part of the world.

## Internet Conferencing

- Internet conferencing allows users to carry on business meetings and seminars, make presentation, provide online education and offer direct customer support.
- Internet conferencing solutions require high speed internet connection at all user sites.

## Electronic Newspaper

- An electronic newspaper is a self-contained, reusable and refreshable version of a traditional newspaper that acquires and holds information electronically.
- Information to be displayed will be downloaded through some wireless internet connections.

## Online Shopping

- It is the process of buying goods and services from merchants who sell on the internet.
- The main components of online shopping are product, selling price, accessibility to people, placement of orders, mode of payments, delivery mechanism.

## 3.5 Different types of Internet connectivity and ISP

### Different types of Internet connectivity

There are different types of connections and speeds to get on the information superhighway.

#### 1) Dial-Up Connections

- A dial up is a method of connecting to the internet using an existing telephone.
  - Dial up connection uses the telephone line to connect to the internet.
  - The modem must dial the telephone every time it wants to connect to the internet hence the name Dial up.
- a) **Modem Dial-Up Connections-**
- The modem connects the computer through the standard phone line which serve as the data transfer medium.
  - A modem changes the digital data from your computer into analog data, a format that can be carried by telephone lines.
- b) **ISDN Dial-Up Connections-**
- This second type of dial up connection is through an ISDN (Integrated services digital network).
  - It is a digital telephone service that can transmit voice, data and control information over an existing single telephone line.

## 2) ADSL Connections

- ADSL (Asymmetric Digital Subscribers Line) connections are becoming more and more widely available and can provide an excellent internet connection.
- The connections work by splitting your phone line into two separate channels, one for data (Internet) and one for voice (phone calls), which means you can talk on the phone and be connected to the internet at the same time.

## 3) Cable Connections

- Cable connections are considered one of the best types of internet connections available to the home user, they offer very fast and reliable connections with a fixed monthly fee.
- A cable connection uses a totally separate medium to transfer data that it doesn't affect your ability to make/receive phone calls.
- Cable connections are always on, eliminating long waits to make a connection.

## ISP (Internet Service Provider)

- When a user initiates a Dialup connection, the modem dials a phone number of an internet service provider (ISP) that is designated to receive Dialup calls.
- The ISP then establishes the connection, which usually takes about tens of seconds and is accompanied by several beeping and buzzing sounds.
- ISPs prefer to the company that provides internet connections to the users.
- Some popular ISPs, are Airtel, MTNL, Vodafone etc.

## Solved Questions

### Short Answer Type Questions.

#### Q1 Define WWW. (2017-Winter)

**Ans.** The World Wide Web, commonly known as the Web, is an information system where documents and other web resources are identified by Uniform Resource Locators, which may be interlinked by hypertext, and are accessible over the Internet.

#### Q2 Define HTTP.

**Ans.** The Hypertext Transfer Protocol is an application layer protocol for distributed, collaborative, hypermedia information systems.

#### Q3 Define a DNS.

**Ans.** The Domain Name System is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities.

#### Q4 Define a FTP. (2016-Winter)

**Ans.** The File Transfer Protocol is a standard network protocol used for the transfer of computer files from a server to a client on a computer network.

## Long Answer Type Questions

### Q.1 Explain e-mail message format? (2017-Winter)

**Ans.** Electronic Mail (e-mail) is one of the most widely used services of the Internet. This service allows an Internet user to send a message in a formatted manner (mail) to other

Internet users in any part of the world. Message in the mail not only contains text, but it also contains images, audio and video data. The person who is sending mail is called sender and person who receives mail is called the recipient. It is just like postal mail service.

#### **Format of E-mail:**

An e-mail consists of three parts that are as follows:

1. Envelope
2. Header
3. Body

These are explained as following below.

#### **1. Envelope:**

The envelope part encapsulates the message. It contains all information that is required for ending any e-mails such as destination address, priority and security level. The envelope is used by MTAs for routing message.

#### **2. Header:**

The header consists of a series of lines. Each header field consists of a single line of ASCII text specifying field name, colon and value. The main header fields related to message transport are:

**To:** It specifies the DNS address of the primary recipient(s).

**CC:** It refers

to carbon copy. It specifies address of secondary recipient(s). **BCC:** It refers to blind carbon copy. It is very similar to Cc. The only difference between Cc and Bcc is that it allows user to send copy to the third party without primary and secondary recipient knowing about this.

**From:** It specifies name of person who wrote message.

**Sender:** It specifies e-mail address of person who has sent message.

**Received:** It refers to identity of sender's, data and also time message was received. It also contains the information which is used to find bugs in routing system. **Return-Path:** It is added by the message transfer agent. This part is used to specify how to get back to the sender.

#### **3. Body:**

The body of a message contains text that is the actual content/message that needs to be sent, such as "Employees who are eligible for the new health care program should contact their supervisors by next Friday if they want to switch." The message body also may include signatures or automatically generated text that is inserted by the sender's email system.

**Q.2 What do you mean by network topologies? What are the major types of network topologies? (2016-Winter) (2015-Winter) (2013-Winter)**

**Ans.**

There are five types of topology in computer networks: These are

1. Mesh Topology
2. Star Topology
3. Bus Topology
4. Ring Topology
5. Hybrid Topology

### **Mesh Topology**

In mesh topology each device is connected to every other device on the network through a dedicated point-to-point link. When we say dedicated it means that the link only carries data for the two connected devices only. Let's say we have  $n$  devices in the network then each device must be connected with  $(n-1)$  devices of the network. Number of links in a mesh topology of  $n$  devices would be  $n(n-1)/2$ .

### **Advantages of Mesh topology**

1. No data traffic issues as there is a dedicated link between two devices which means the link is only available for those two devices.
2. Mesh topology is reliable and robust as failure of one link doesn't affect other links and the communication between other devices on the network.
3. Mesh topology is secure because there is a point-to-point link thus unauthorized access is not possible.
4. Fault detection is easy.

### **Disadvantages of Mesh topology**

1. Amount of wires required to connect each system is tedious and headache.
2. Since each device needs to be connected with other devices, number of I/O ports required must be huge.
3. Scalability issues because a device cannot be connected with a large number of devices with a dedicated point-to-point link.

### **Star Topology**

In star topology each device in the network is connected to a central device called hub. Unlike Mesh topology, star topology doesn't allow direct communication between devices; a device must have to communicate through hub. If one device wants to send data to other device, it has to first send the data to hub and then the hub transmits that data to the designated device.

### **Advantages of Star topology**

1. Less expensive because each device only needs one I/O port and needs to be connected with hub with one link.
2. Easy to install
3. Less amount of cables required because each device needs to be connected with the hub only.
4. Robust, if one link fails, other links will work just fine.
5. Easy fault detection because the link can be easily identified.

### **Disadvantages of Star topology**

1. If hub goes down everything goes down, none of the devices can work without hub.
2. Hub requires more resources and regular maintenance because it is the central system of star topology.

### **Bus Topology**

In bus topology there is a main cable and all the devices are connected to this main cable through drop lines. There is a device called tap that connects the drop line to the main cable. Since all the data is transmitted over the main cable, there is a limit of drop lines and the distance a main cable can have.

### **Advantages of bus topology**

1. Easy installation, each cable needs to be connected with backbone cable.
2. Less cables required than Mesh and star topology

### **Disadvantages of bus topology**

1. Difficultly in fault detection.
2. Not scalable as there is a limit of how many nodes you can connect with backbone cable.

### **Ring Topology**

In ring topology each device is connected with the two devices on either side of it. There are two dedicated point-to-point links a device has with the devices on the either side of it. This structure forms a ring thus it is known as ring topology. If a device wants to send data to another device then it sends the data in one direction, each device in ring topology has a repeater, if the received data is intended for other device then repeater forwards this data until the intended device receives it.

### **Advantages of Ring Topology**

1. Easy to install.
2. Managing is easier as to add or remove a device from the topology only two links are required to be changed.

### **Disadvantages of Ring Topology**

1. A link failure can fail the entire network as the signal will not travel forward due to failure.
2. Data traffic issues, since all the data is circulating in a ring.

### **Hybrid topology**

A combination of two or more topology is known as hybrid topology. For example a combination of star and mesh topology is known as hybrid topology.

### **Advantages of Hybrid topology**

1. We can choose the topology based on the requirement for example, scalability is our concern then we can use star topology instead of bus technology.
2. Scalable as we can further connect other computer networks with the existing networks with different topologies.

### **Disadvantages of Hybrid topology**

1. Fault detection is difficult.
2. Installation is difficult.
3. Design is complex so maintenance is high thus expensive.

## **EXERCISE**

### **ShortAnswerTypeQuestions.**

- Q1** Define computernetwork.(2014-Summer)
- Q2 Whatarethevarioustypes ofnetworks?
- Q3 What isaMAN?
- Q4** Whatarethedifferenttopologiesusedincomputernetwork?(2018-Summer)
- Q5 Name some Internet service providersinIndia?
- Q6 What doyoumeanby awebpage?
- Q7 What isawebsite?
- Q8 Whatise-mail?Whataretheusesofe-mail?
- Q9 Whatare the benefitsofe-mail?
- Q10 Whatarethelimitationsofe-mail?
- Q11** DefineaURL.(2013-Summer)
- Q12** WhatdoyoumeanbyNetwork topology?(2016-Winter)
- Q13 Whataretheadvantages and DisadvantagesofBusTopology?
- Q14 What isamodem?

### **LongAnswerTypeQuestions**

- Q1 Discuss about the various categories of computer network. Give a comparisonsbetweenLANandWAN.
- Q2** Describevariouscategoriesof networkwith example.(2016-Summer)

\*\*\*\*\*

# CHAPTER-4:FILEMANAGEMENT ANDDATAPROCESSING

## 4.1 Concept of File and Folder

### Concept of file and folder:-

- In earlier DOS based system, we had organized data into files and directories.
- In GUI based operating system, such as window, we have file and folders, in which data are reorganized during storage in computer memory.
- However, the unit of raw data in binary format is either byte (B) or kilobyte (KB) or megabyte (MB) or gigabyte (GB).
- A byte is the smallest unit of information. It is used to measure the size of documents.  $1 \text{ Byte} = 8 \text{ bits}$   
 $1 \text{ KB} = 2^{10} \text{ Bytes} = 1024 \text{ Bytes}$   
 $1 \text{ MB} = 2^{10} \text{ KB} = 1024 \text{ KB}$   
 $1 \text{ GB} = 2^{10} \text{ MB} = 1024 \text{ MB}$   
 $1 \text{ TB} = 2^{10} \text{ GB} = 1024 \text{ GB}$

### Files:

- Files are the most basic unit of data that a user can store on a disk. A file is the common storage unit in a computer.
- All program and data are contained in a file and the computer needs and writes files.
- In every program, image, video, song and document are stored in a file.
- It is possible to move a file from one folder to another.
- One can create, save, open, move and delete files.
- There are different types of files depending on the type of information they contain. There are image files etc.
- The files are assigned a type of file which can be known from the extension of the filename.
- The file name can have up to 255 characters, it can contain letters, numbers, blank spaces and special characters like dashes, underlines, etc. but there is a group that cannot be used ( $\backslash$ , /, >, <, |)
- File extension files are identified by a short "extension" at the end of their name.
- Ex: Soumya.jpg is a JPEG image
- Chandan.doc is a Microsoft Word doc
- ABC.exe is an executable application in windows.

### Folders:

- A folder is a collection of multiple files.
- A folder holds one or more files and it can be empty with just a name.
- A folder can also store other folders called subfolders.
- Folders were also called "directories" in operating systems before windows.
- It would become almost impossible to manage hundreds of files in your computer.

### Different between a file and a folder:

#### File:

- Files store data, whether text, music or item.
- Files are taking spaces on computer memory.
- Each file has its own extension.
- Easily move or copy data from one file to another.
- Cannot create any folder or subfolder within a file.

### **Folder:**

- A folder stores files and other folders.
- Folders are not taking space on computer memory.
- Folders do not have any extension.
- Copy or move files from one folder to another folder.
- Can create different types of files or subfolders in a folder.

## **4.2 File Access and Storage methods. Sequential, Direct, ISAM**

### **File Access and storage method:**

- An access method defines the technique that is used to store and receive data.
- An access method is a function of a main frame operating system that enables access to data on a disk or other external device.
- In computing an access method is a program or a hardware mechanism that moves the data between the computer and an outputting device such as a hard disk or a display terminal.
- It is also used to describe the way that data is located within a large unit of data such as a dataset or files.
- There are various types of access methods

#### **1) Random Access or Direct Access:**

- Direct access method is based on a disk model if a file, since disks allow random access to any file block.
- This type of access method provides a speedy access to the file. It provides immediate access to a large amount of information.
- It allows the program to read and write the records in a rapid manner in non-particular orders.
- For direct access, we can view the file as a number sequence of blocks or records.
- This method is usually used in a database.

#### **2) Sequential Access:**

- This is the most common method.
- Here the information present in the file is accessed in a sequential fashion, one record after the other.
- It is a very common approach which is used by editors and compilers usually.
- The Read and write operation from the major part of the operations done on a file.
- A read operation reads the next portion of the file and automatically advances the file pointer, which tracks the I/O locations.
- A write operation appends to the end of files and advances to the end, if the newly written material.

#### **3) Indexed Sequential Access Method (ISAM)-**

- This method is built on top of direct access method.
- Here an index contains the pointers to various blocks of the file.
- So, to find a record inside a file, we first search the index and later use the pointer obtained to access the file directly and find the record we have been searching for.
- The records of the data file are stored in sequential order according to some data attributes.
- Since ISAM is static, it doesn't change its structure if records are added or deleted from the data file.
- ISAM is available in many variations on microcomputer, minicomputers, and mainframe computers.



## 4.3 Data Capture, Data Storage, Data Processing and Retrieval

### 4.3.1 Data Capture: -

- Data capture is the process of identification and extraction of data from a scanned document, often to be sent to a workflow for routing and action as part of a business process.
- Multiple methods are available for capturing data from unstructured documents (letters, invoices, email, fax, forms etc.)
- Methods of capture from documents in electronic format are identified below:

#### Singleclick:

- It is an optical character recognition (OCR) tool that can be used to capture machine-produced characters in low volume ad-hoc capture application and populating a line of business application.

#### OCR:

- OCR technology provides the ability to successfully capture machine-produced characters in full page.
- OCR systems can recognize many different OCR fonts as well as typewriter and computer printed characters.

#### ICR (Intelligent Character Recognition):

- ICR is the computer translation of hand-printed and written characters.
- Data is entered from hand-printed forms through a scanner and the image of the captured data is then analyzed and then translated by sophisticated ICR software.

#### Barcode Recognition:

- Dependent upon the type of barcode that is used, the amount of metadata that can be included is high, as is the level of recognition.
- The application of single and multiple barcodes to particular document types such as proof of delivery notes.

#### IDR (Intelligent Document Recognition):

- The level of capability is dependent upon the individual product.
- These applications are used to capture metadata from documents that is rule-based.
- Ex: The product will identify post codes, logos, keywords.

### 4.3.2 Data Storage:

- Data storage is the holding of data in an electromagnetic form for access by a computer processor.
- There are two kinds of storage:

- a) Primary storage is data that is held in RAM and other memory devices that are built into computers.
- b) Secondary storage is data that is stored on external storage devices such as hard disks, tapes, CD, Pen drive etc.

Following are some main devices for data storage:

- Hard disks
- Floppy disks
- Optical disks
- CD
- Pen drives
- Flat memory card/memory card

### 4.3.3 Data Processing:

- Data processing must be processed in order to convert it into information.
- For this purpose, different operations may be performed on data.
- Data processing is defined as a sequence of operations on data to convert it into useful information.
- The data processing can be accomplished through following methods:

#### 1. Manual Data Processing:

- In this method, data is processed manually without using any machine or tool to get required results.
- In manual data processing, all the calculations and logical operations are performed manually on the data.
- Ex: Mark sheets, fee receipts

#### 2. Mechanical Data Processing:

- In this method, data is processed by using different devices like typewriters, mechanical printers or other mechanical devices.
- Examination board and printing press use mechanical data processing devices frequently.

#### 3. Electronic Data Processing:

- It is the modern technique to process data.
- The data is processed through computer; data and set of instructions are given to the computer as input and the data according to the given set of instructions.
- The computer is also known as electronic data processing machine.
- Ex: results of students are prepared through computers.

### 4.3.4 Data Retrieval:

- Data is one of the most important assets of any business.
- Data recovery refers to the whole process of salvaging this lost data that is corrupted, failed, damaged or inaccessible.
- Lost files can occur because of any of the below possibilities.

- 1) File was mistakenly deleted.
- 2) File was corrupted or deleted by scan disk.
- 3) Another program deleted the file.
- 4) File is password protected.

- Following are some different methods of data recovery:-

#### 1) Physical damage to storage devices:

- Different failure can cause physical damage to your storage media.

#### 2) Media errors and corrupt partitions and file systems:

- In some cases, media errors or damage to the file system or partition table can make the data on a hard drive to be unreadable.

#### 3) Online data recovery:

- This is another popular method of data recovery. Sydney business users restore deleted or lost files.

- It is a method of data recovery that is performed over the internet without necessarily having the computer or the drive in possession.

## Solved Questions

### Short Answer Type Questions.

**Q.1 What is the difference between file and folder.**

**Ans:-.**

Sl. No.	Key	File	Folder
1	Extension	Files may or may not have extensions.	Folders do not have extensions.
2	Container	A File can not contain another file/folder.	A folder can contain any number of file/folders.
3	Memory size	A file has certain size and memory consumption.	A folder has no size of its own. It derives the size from the files it contains.
4	Attributes	Name, Extension, Date, Time, Length and Protection (Read-Only, hidden etc.)	Name, Date, Time and Protection (Read-Only, hidden etc.)

## Q.2 What do you mean by ISAM? (2016-Winter)

Ans:-i.

- I. It stands for Indexed Sequential Access Method.
- II. ISAM is a method for creating, maintaining, and manipulating computer files of data so that records can be retrieved sequentially or randomly by one or more keys.
- III. In this method, each record has the address of its data block, searching a record in a huge data base is quick and easy.
- IV. This method supports range retrieval and partial retrieval of records.

## Q.3 Define OCR.

Ans

- :-
- I. Stands for "Optical Character Recognition.
  - II. OCR is a technology that recognizes text within a digital image.
  - III. It is commonly used to recognize text in scanned documents, but it serves many other purposes as well.

## Q. 4 What is folder. (2017-Winter)

Ans:- A Folder is a collection of tables, charts, or other outputs in the Report tree.

- It is the same idea as having folders on your computer.
- They are created by right-clicking on the Report tree and selecting Add Folder.
- Folders are also called directories because of the way they organize data within the file system of a storage device.

## Long Answer Type Questions.

### Q.1 What is file access? Explain the various types of file access method. (2014-Winter)(2015-Winter)

Ans:- A file access definition can control access to data in specified tables and columns, or to tables and columns for which access is not granted explicitly. You define access permissions by creating an access list for a table, column, or the default.

When a file is used, information is read and accessed into computer memory and there are several ways to access this information of the file. Some systems provide only one access method for files.

There are three ways to access a file into a computer system: Sequential-Access, Direct Access, Index sequential Method.

#### 1. **Sequential Access-**

Data is accessed one record right after another record in an order. When we use read command, it moves ahead pointer by one. When we use write command, it will allocate memory and move the pointer to the end of the file. Such a method is reasonable for tape.

**2. DirectAccess–**

Another method is direct access method also known as relative access method. Afiled- length logical record that allows the program to read and write record rapidly.in no particular order. The direct access is based on the disk model of a file sincedisk allows random access to any file block. For direct access, the file is viewed asa numbered sequence of block or record. Thus, we may read block 14 then block59 and then we can write block 17. There is no restriction on the order of readingandwritingfordirectaccessfile.

**3. Indexsequentialmethod–**

It is the other method of accessing a file which is built on the top of the sequentialaccess method. These methods construct an index for the file. The index, like anindex in the back of a book, contains the pointer to the various blocks. To find arecord in the file, we first search the index and then by the help of pointer weaccessthefiledirectly.

**Q. 2Discussaboutthe Data storedevices.**

**Ans:-**Astoragedeviceisapieceofcomputerhardwareusedforsaving,carryingandpulling out data. It can keep and retain information short-term or long-term. It can be adevice inside or outside a computer or server. Other terms for storage device isstoragemediumorstoragemedia.

A storage device is one of the basic elements of any computer device. It almostsaves all data and applications in a computer except for hardware firmware. Itcomesindifferentshapesandsizes dependingon theneeds andfunctionalities.

Therearetwo differenttypes ofstoragedevices:

	<b>PrimaryStorageDevice</b>	<b>SecondaryStorage Device</b>
<b>Size</b>	Smaller	Larger
<b>DataRetention</b>	Temporary	Permanent
<b>Location</b>	Internal	Internal/External
<b>E.g</b>	RAM,ROM	Magnetic Storage Device,Floppy diskette, Hard drive,Magnetic strip,Cassettetape,etc.

**EXERCISE**

**ShortAnswerTypeQuestions.**

Q1 What is the difference between Random Access method and sequential accessmethod?  
*(2017-Winter)*

Q2 Whatis Dataprocessing anddiscussaboutsosomeoperations thatcanbepreformedondata?

**LongAnswerTypeQuestions.**

Q1Whatdo youmeanbyfileaccess?Explainthevarious types offileaccesstechniques.*(2017-Summer)*

Q2 Discussabout the DataRetrieval.

# CHAPTER-5:PROBLEMSOLVINGMETHODOLOGY

## Problemsolving

Solving problems is the core of computer science. Programmers must first understand how a human solves a problem, then understand how to translate this "algorithm" into something a computer can do, and finally how to "write" the specific syntax (required by a computer) to get the job done. It is sometimes the case that a machine will solve a problem in a completely different way than a human.

Computer Programmers are problem solvers. In order to solve a problem on a computer you must:

1. Know how to represent the information (data) describing the problem.
2. Determine the steps to transform the information from one representation into another

## 5.1 Algorithm, Pseudocode and Flowchart

### 5.1.1 Algorithm

An algorithm is a set of specific steps to solve a problem. Think of it this way: if you were to tell your 3 year old niece to play your favorite song on the piano (assuming the niece has never played a piano), you would have to tell her where the piano was, and how to sit on the bench, and how to open the cover, and which keys to press, and which chord to press them in, etc, etc, etc.

#### Definition:

- An algorithm is a well-defined procedure that allows a computer to solve a problem.
- An algorithm is defined as the step-by-step solution of a problem in user's language.
- It is considered as an effective procedure for solving a problem in a finite number of steps.
- Another way to describe an algorithm is a sequence of unambiguous instructions.
- In fact, it is difficult to think of a task performed by your computer that does not use algorithms.

The characteristics of Algorithms are

- Precise
- Unambiguous
- Finite termination
- Unique solution

Example:

1. Algorithm to find out sum of two numbers to be taken as input.

```
Step-1 Read the 1st number
x
Step-2 Read the 2nd
number y
Step-3 Sum = x + y
Step-4 Print Sum
```

**This is an example where only sequence is exhibited**

2. Algorithm to find out larger between two numbers to be taken as input.

```
Step-1 Read the 1st number
x
Step-
2 Read the 2nd number y
Step-
3 If x > y
Then Print
x
Else if
x < y Then P
```

```
rinty  
ElsePrint BothareEqual
```

ThisisanexamplewhereBranching is exhibited

3. Algorithmtofind outsumof  
first10naturalnumbers.Step-1i=1,Sum=0  
Step-2 Repeat step 3 and 4 while  
i<>10Step-3Sum=Sum+i  
Step-4i=i+1Step-5 PrintSum

Thisisanexample where Repetitionisexhibited

## 5.1.2 Pseudocode

ItisaconcisedescriptionalgorithminEnglishlanguage thatusesprogramminglanguageconstructs.Itcontainsoutlinesoftheprogramthatcanbeeasilyconvertedtoprogram.Itfocusesonthelogicofthealgorithmwithoutgivingstressonthesyntaxofprogramminglanguage.Thisismeantforunderstandingthelogicoftheprogrameasily.Flowchartcanbeconsideredasanalternativetopseudocode. Severalconstructs/keywordsofprogramminglanguagecan be usedin thealgorithmtowritethe pseudocode.

Someofthemare

```
If...Endif  
Do while ... end  
doWhile ... end  
whileRepeat...  
until  
For...endforC  
aseendcase  
Call  
Return
```

## 5.1.3 Flowchart

Flowchart is a graphical or symbolic representation of the process of solution to aproblem or algorithm. It helps to visualize the complex logic of the solution of the problem in asimplifiedmannerthroughdiagrammaticrepresentation.Eachstepofthealgorithmispresented using a symbol and a short description. The different symbols used for the flowchartare

Symbol	Purpose	Description
	Flowline	Indicates the flow of logic by connecting symbols.
	Terminal(Stop/Start)	Represents the start and the end of a flowchart.
	Input/output	Used for input and output operation.
	Processing	Used for arithmetic operations and data manipulations.
	Decision	Used for decision making between two or more alternatives.
	On-page Connector	Used to join different flowlines.
	Off-page Connector	Used to connect the flowchart portion on a different page.
	Predefined Process/Function	Represents a group of statements performing one processing task.

## 5.2 Generation of Programming Languages

### Programming Language

Programming language is a tool to express the logic or instructions for understanding of the computer. Any programming language has two components:

- a) Syntax
- b) Semantics

Syntax refers to the rules to be followed for writing valid program statements. Compiler can detect errors in syntax while compiling the program. Semantics is associated with logic of the program. Compiler cannot detect these semantic errors. The user or programmer can diagnose semantic errors.

There are good numbers of High level languages, each meant for specific area of data processing. Commonly known languages are BASIC, FORTRAN, COBOL, Pascal, C, C++ etc. While FORTRAN is good for Numerical and scientific calculation, COBOL is good for Business applications involving large amount of data handling.

### Generation of Programming Language

The Programming languages can be classified into 4 generations:

- 1st Generation: Machine Language
- 2nd Generation: Assembly Language
- 3rd Generation: High Level Language
- 4th Generation: Very High Level Language



Machine Level language contains instructions in binary form i.e. in 0s and 1s. Thus writing instruction was very difficult and needs heavy expertise. This was used in early days computers.

Assembly level language instructions were written using symbolic codes known as mnemonics. In comparison to Machine language, it is relatively easier to write program, but still it requires lot of expertise. A translator called assembler is used to translate assembly language program to machine level language.

High level language contains instructions in English like words so that user will find it easier to formulate and write the logical statements of the program. Here the logic may spread over multiple statements as against a single statement in assembly language. It uses a translator or called compiler for translation of High level language program to machine level language program. There are many High level languages used for programming such as BASIC, FORTRAN, COBOL, PASCAL, C, C++ etc.

Very High Level language otherwise called as 4GL uses non-procedural logical statements. A typical example of 4GL is the query languages such as SQL.

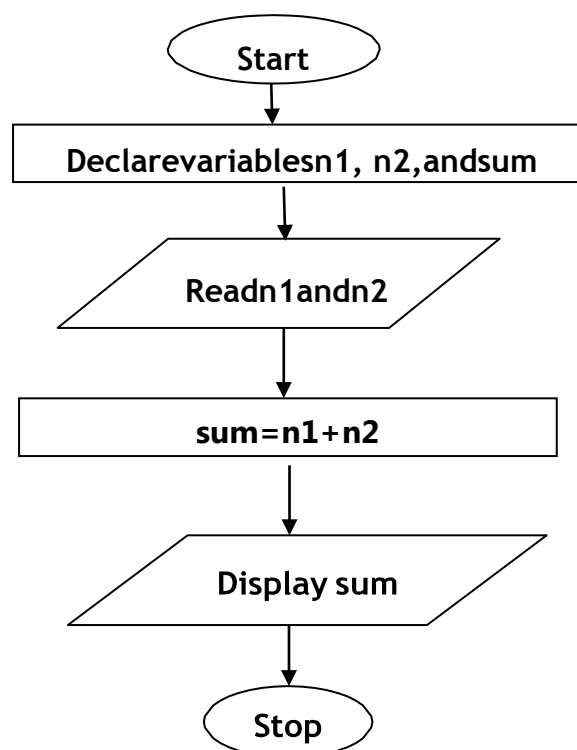
### 5.3 Structured Programming Language

Structured Programming is also known as Modular Programming. In this type of programming technique, the program shall be broken into several modules. This helps in managing memory efficiently as the required module of the program will be loaded into the memory only and not the entire program. This will also enhance code reuse. Writing, understanding, debugging and modifying the individual module of the program is also easier.

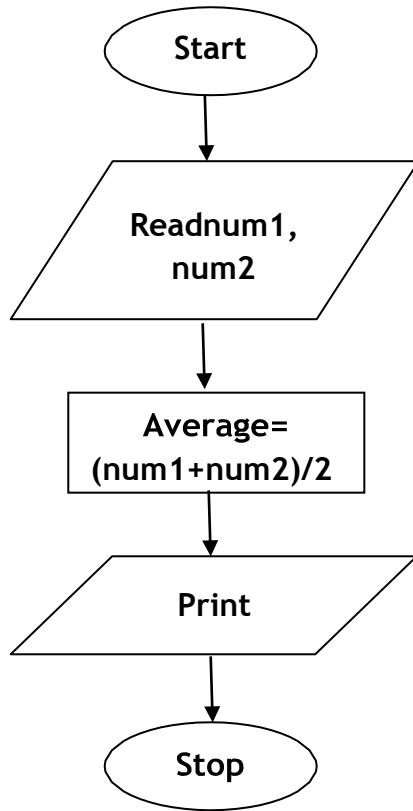
### 5.4 Examples of Problem Solving through Flowchart

#### Example

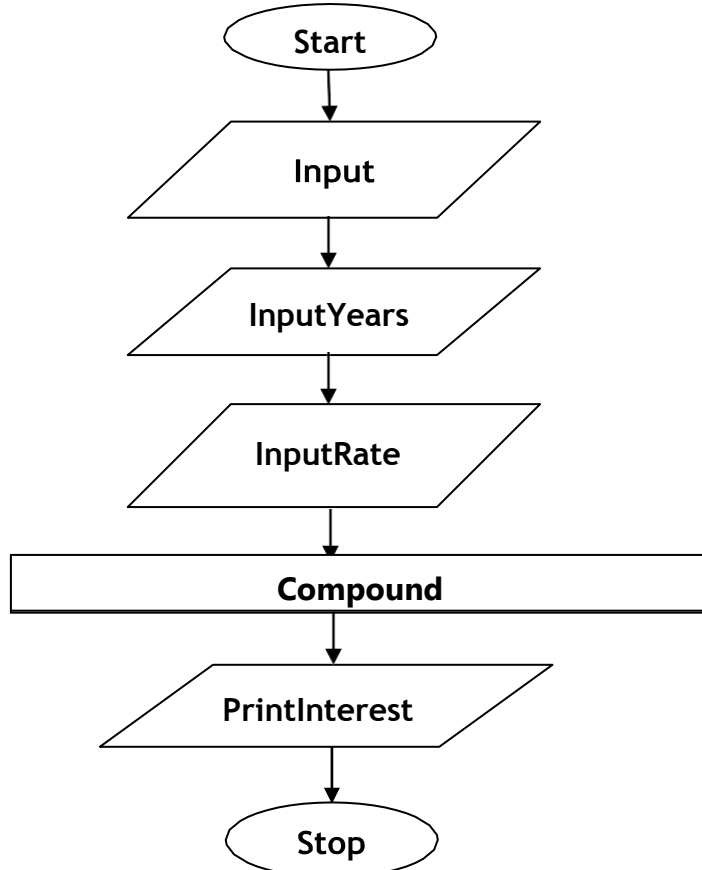
1. Add two numbers entered by the user.



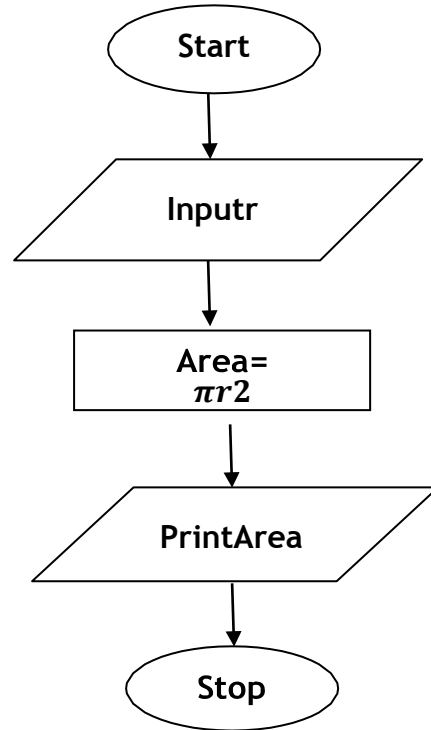
## 2. Flowchart to calculate the average of two numbers



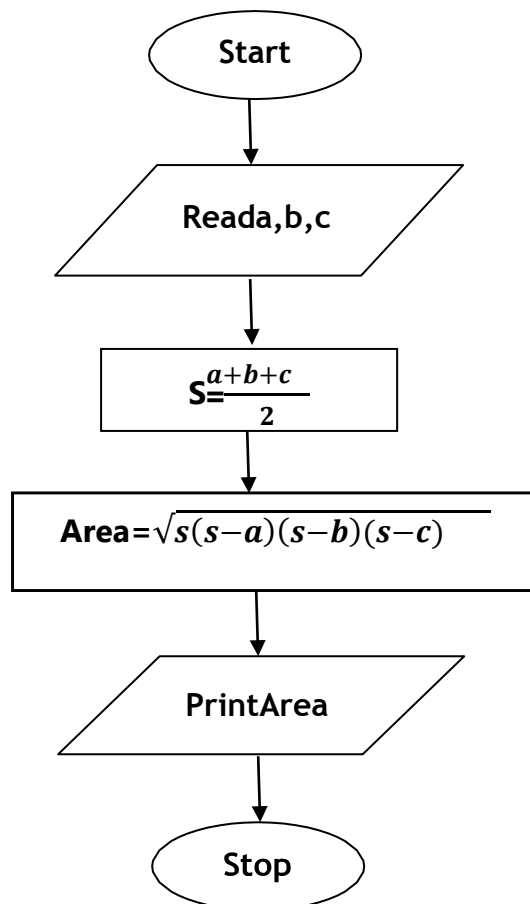
## 3. Flowchart to Calculate the Interest of a Bank Deposit



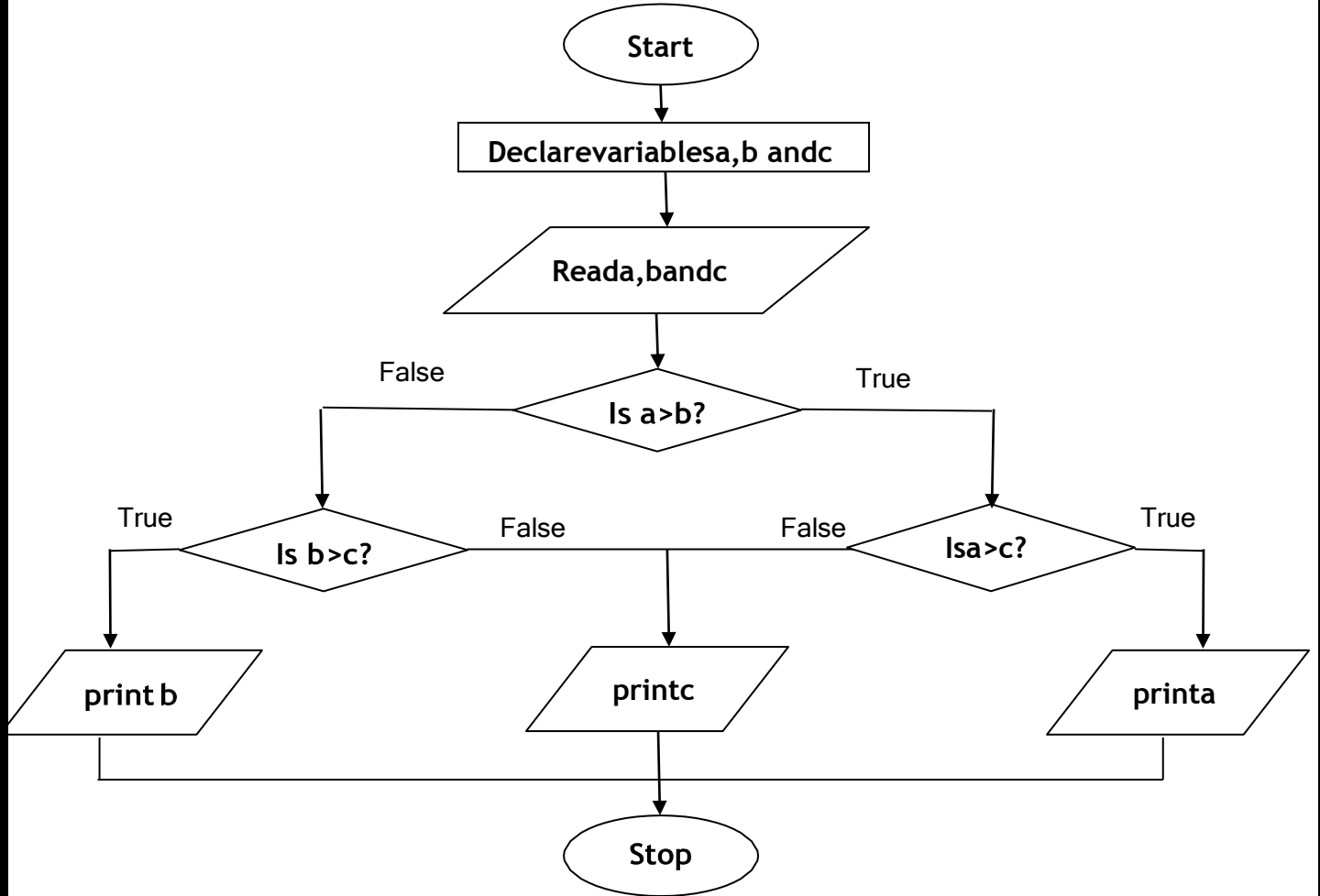
4. Flowchart to calculate the area of a circle.



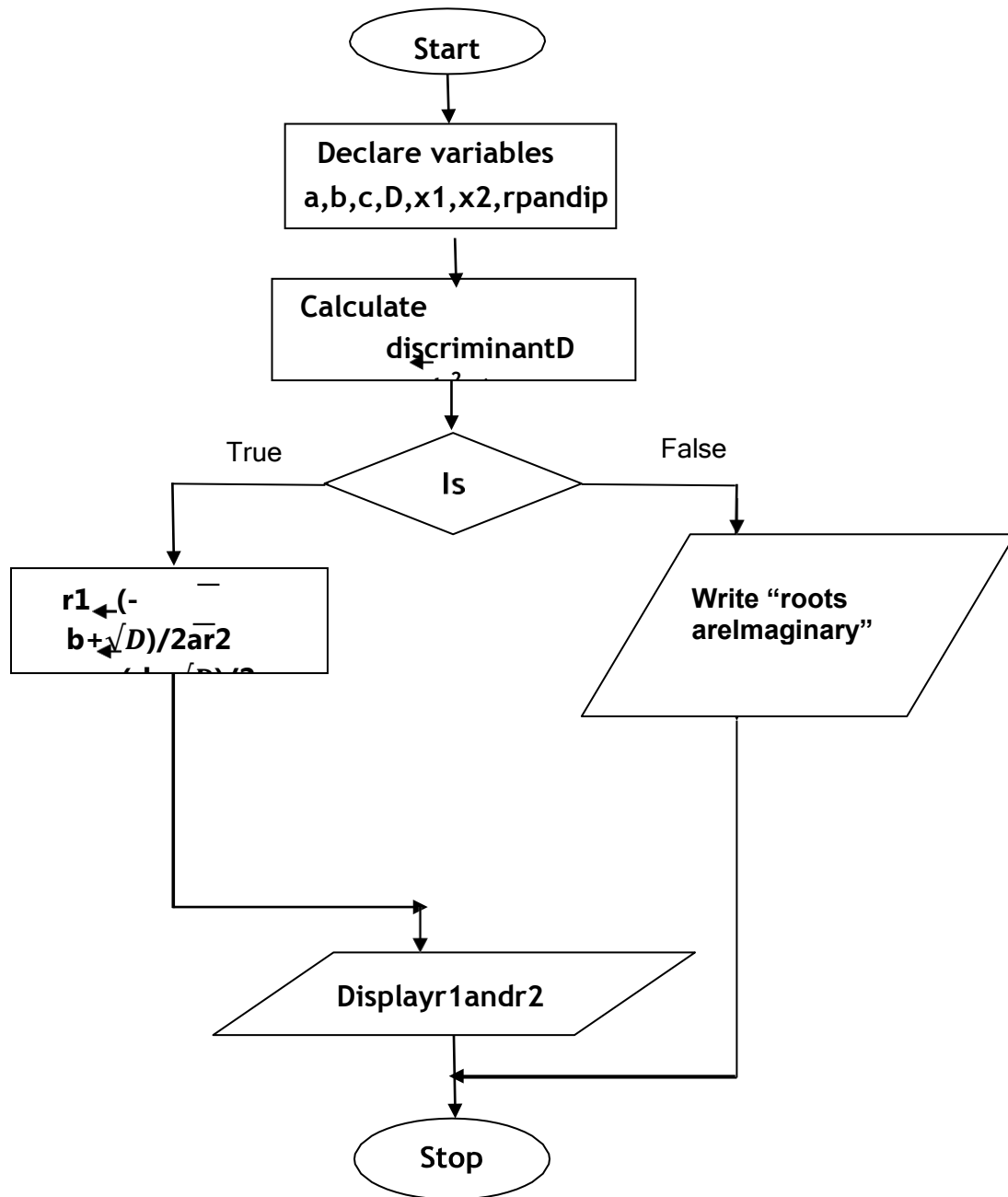
5. Flowchart to calculate the area of a triangle.



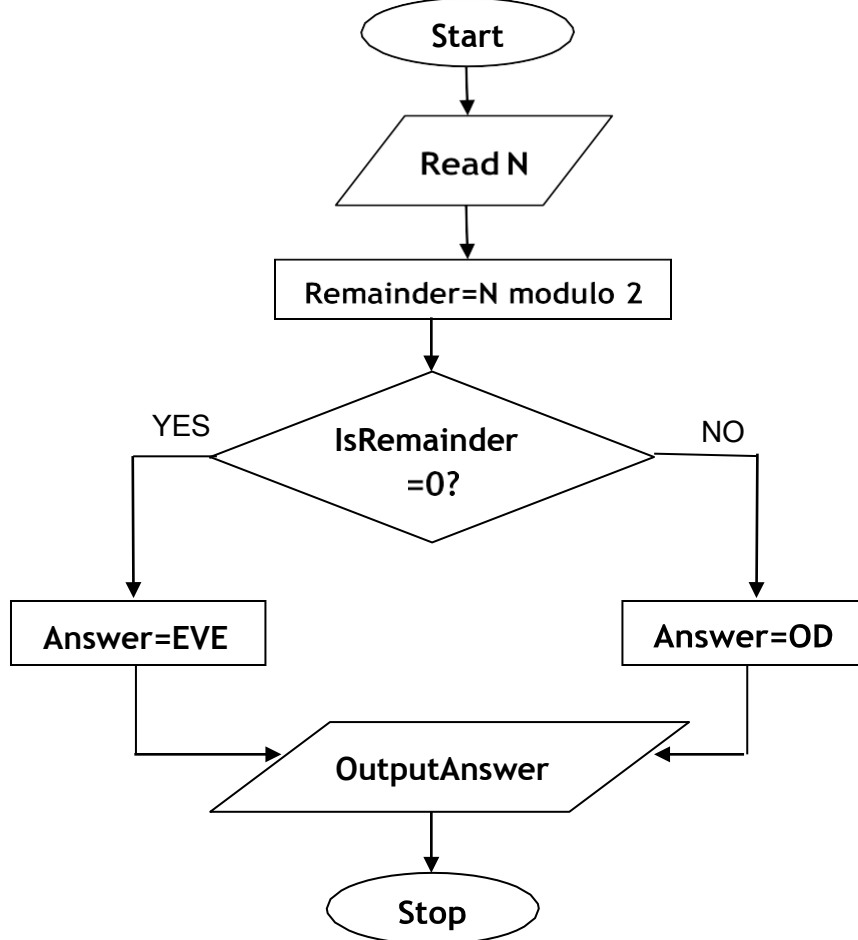
6. Find the largest among three different numbers entered by the user.



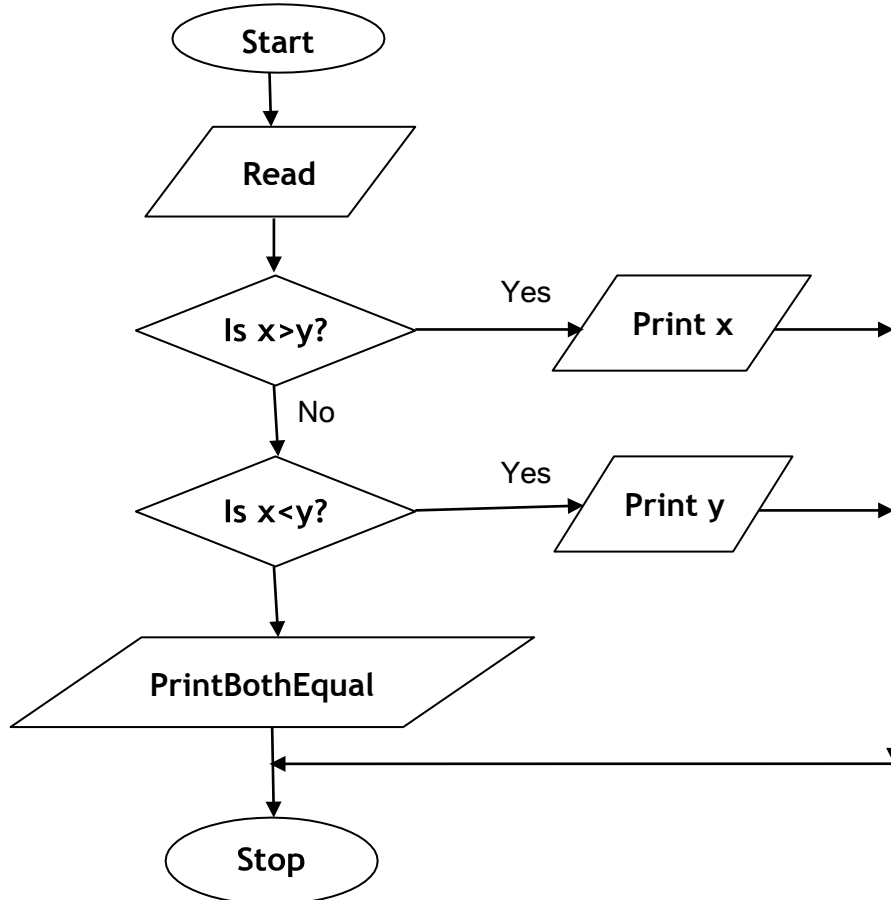
7. Find all the roots of a quadratic equation  $ax^2+bx+c=0$



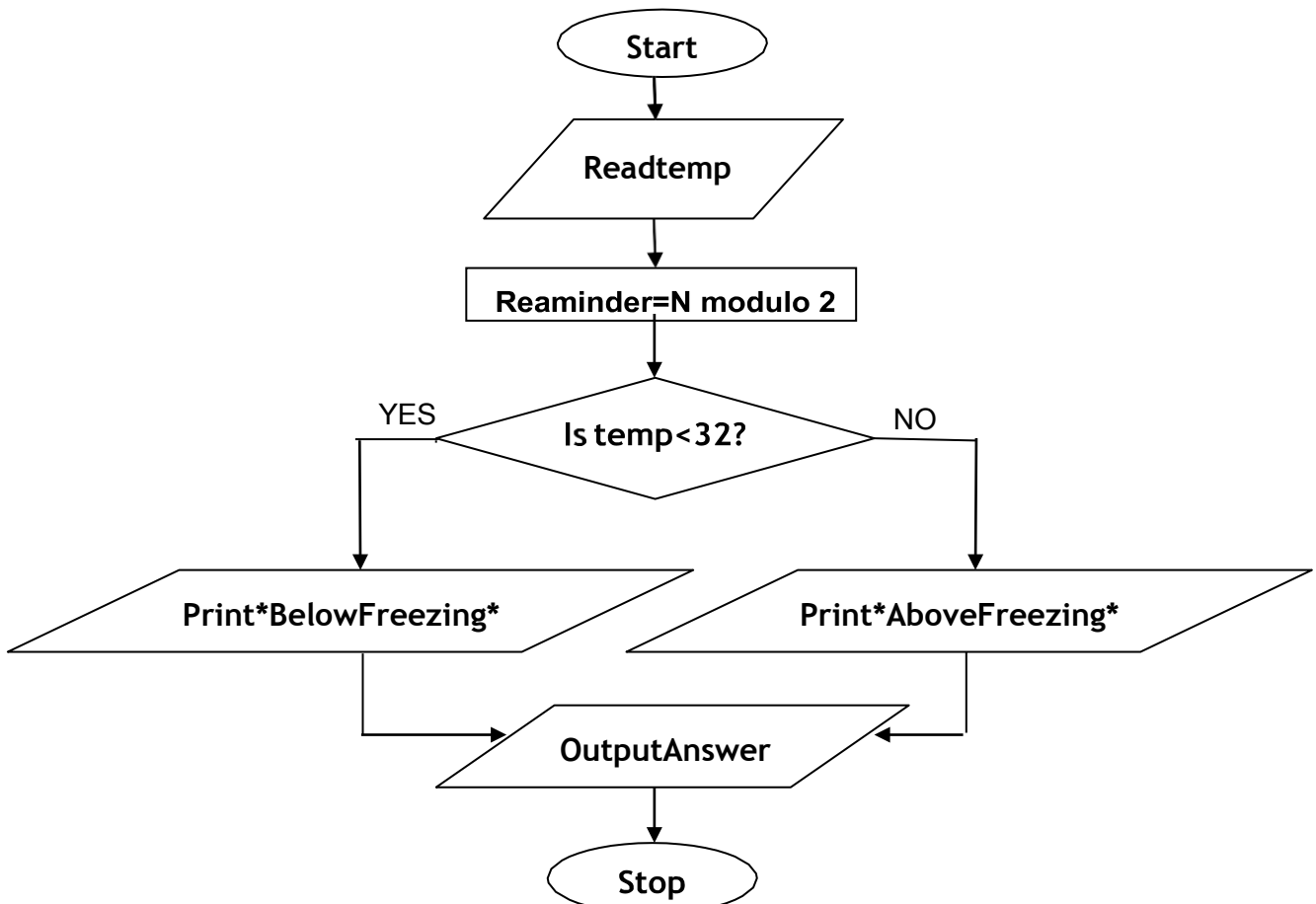
8. Flowchart to Determine and Output Whether Number is Even or Odd



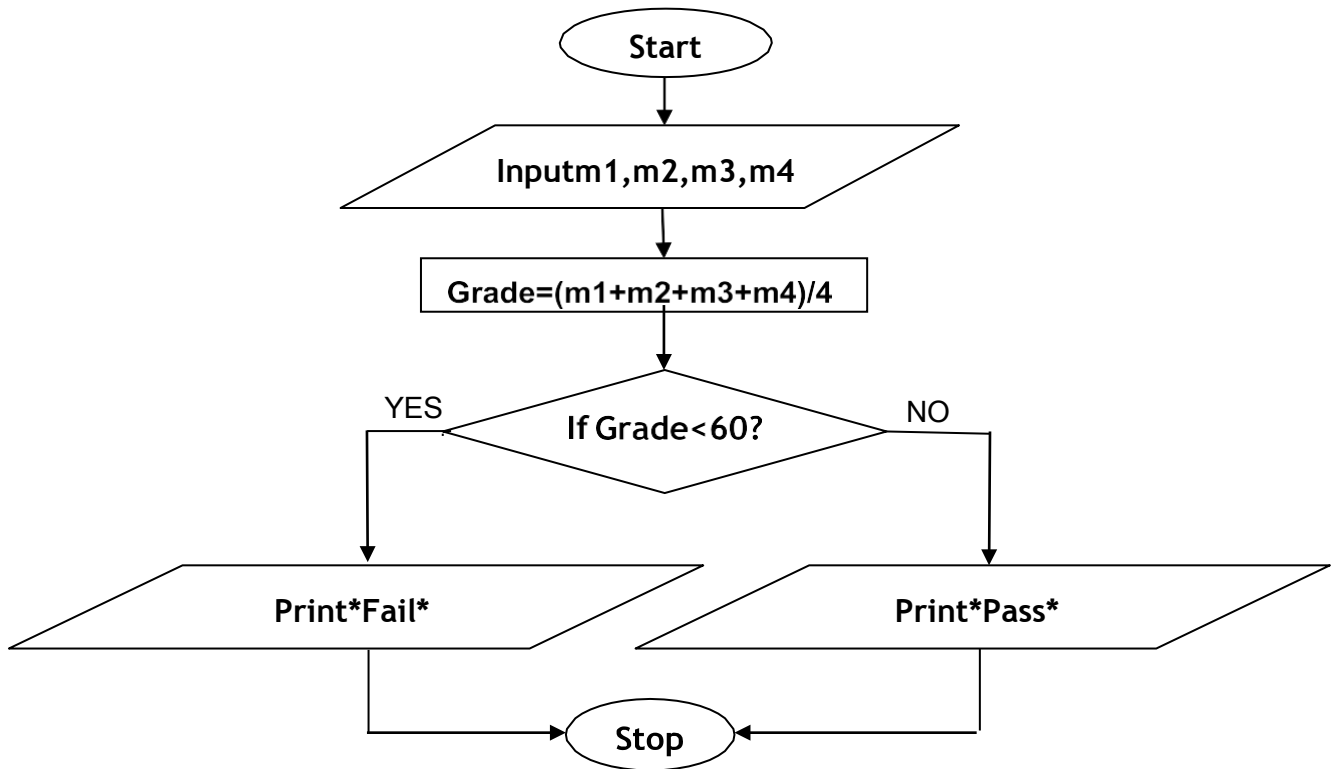
9. Flowchart to find out larger between two numbers to be taken as input



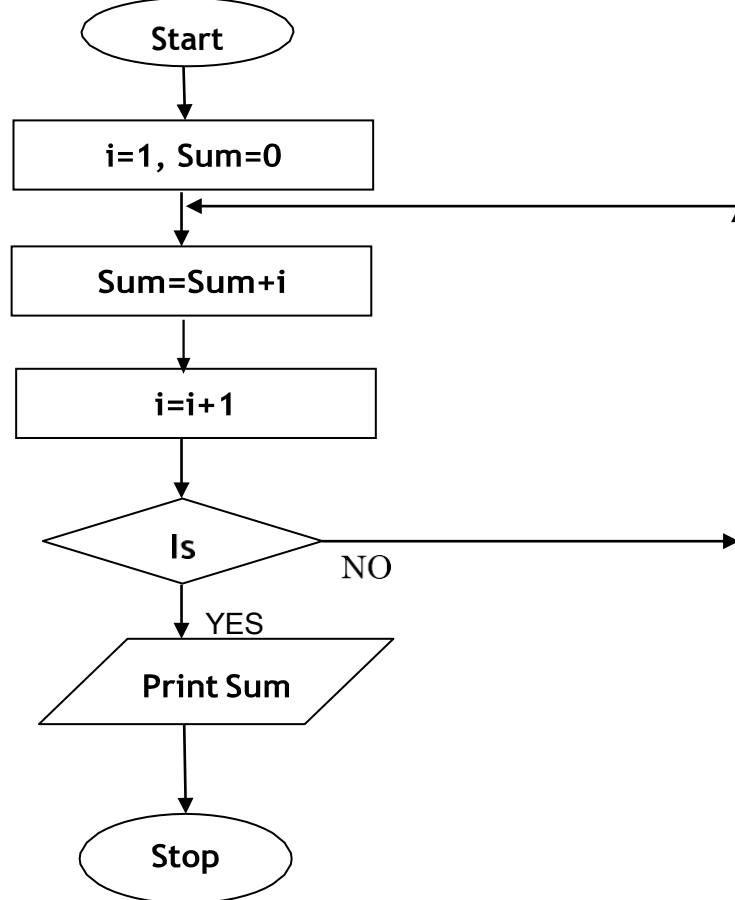
10. Flowchart to Determine Whether a Temperature is Below or Above the Freezing Point



11. Flowchart to Determine Whether a Student Passed the Exam or Not



## 12. Flowchart to find out sum of first 10 natural numbers



## Solved Questions

### Short Answer Type Questions.

#### Q.1 Define an algorithm. (2013-Winter)

**Ans:-** An algorithm is a set of instructions designed to perform a specific task. This can be a simple process, such as multiplying two numbers, or a complex operation, such as playing a compressed video file. Algorithm is defined as the step-by-step solution of a problem in user's language. It is considered as an effective procedure for solving a problem in a finite number of steps.

#### Q.2 What are the characteristics of an Algorithm?

**Ans:-**

##### 1. Input specified:

The input is the data to be transformed during the computation to produce the output. An algorithm should have 0 or more well-defined inputs.

##### 2. Output specified:

The output is the data resulting from the computation (your intended result). An algorithm should have 1 or more well-defined outputs, and should match the desired output.

##### 3. Effectiveness:

For an algorithm to be effective, it means that all those steps that are required to get to the output must be feasible with the available resources.



#### 4. Independent:

An algorithm should have step-by-step directions, which should be independent of any programming code. It should be such that it could be run on any of the programming languages.

#### Q.3 What is the function of a Flowchart? What information does it depict?

Ans:-

- A flowchart describes the steps software takes to process information, from the beginning data inputs, through processing and logical decisions, to the point where the program ends.
- Software developers use flowcharts to plan out how computer applications work before programmers write the code.
- It helps to visualize the complex logic of the solution of the problem in a simplified manner through diagrammatic representation.
- Each step of the algorithm is presented using a symbol and a short description.

What information it depicts:-

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.

#### Q.4 Differentiate between an algorithm and flowchart (2015-

Winter) Ans:-

S.NO	Algorithm	Flowchart
1.	Algorithm is step by step procedure to solve the problem.	Flowchart is a diagram created by different shapes to show the flow of data.
2.	Algorithm is complex to understand.	Flowchart is easy to understand.
3.	In algorithm plain text are used.	In flowchart, symbols/shapes are used.
4.	Algorithm is easy to debug.	Flowchart is hard to debug.
5.	Algorithm is difficult to construct.	Flowchart is simple to construct.

## LongAnswerTypeQuestions.

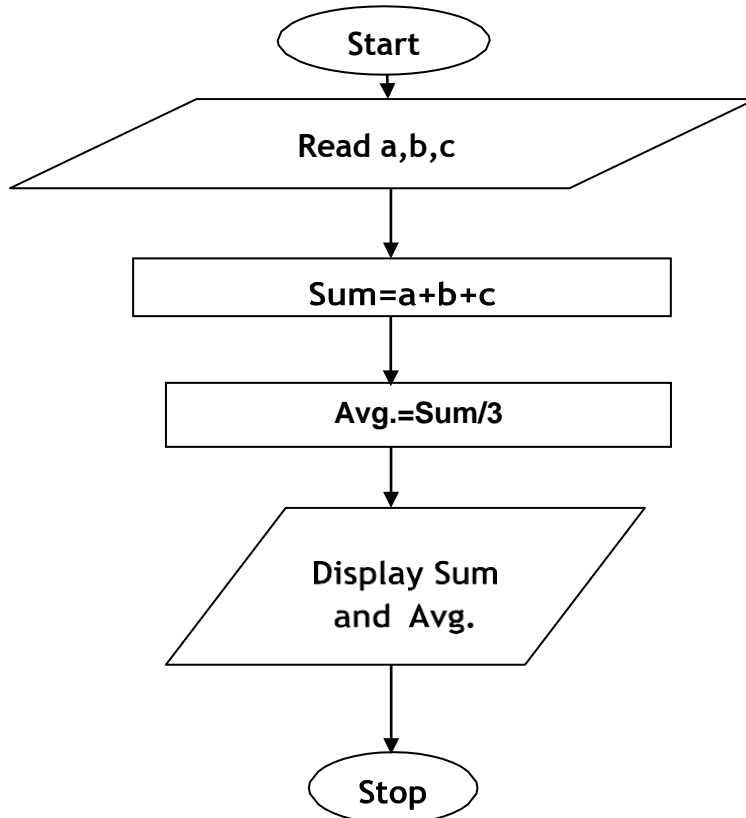
Q.1 Write algorithms and flowchart of the followings:

a. To find sum & average of 3 numbers. A

ns:-Algorithm

- Step 1: Start.
- Step 2: Read the three numbers suppose "a", "b", "c" from the user.
- Step 3:  $\text{Sum} = a + b + c$ ;
- Step 4:  $\text{Avg.} = \text{Sum} / 3$ .
- Step 5: Display "Sum" and "Avg".
- Step 6: End.

Flowchart:

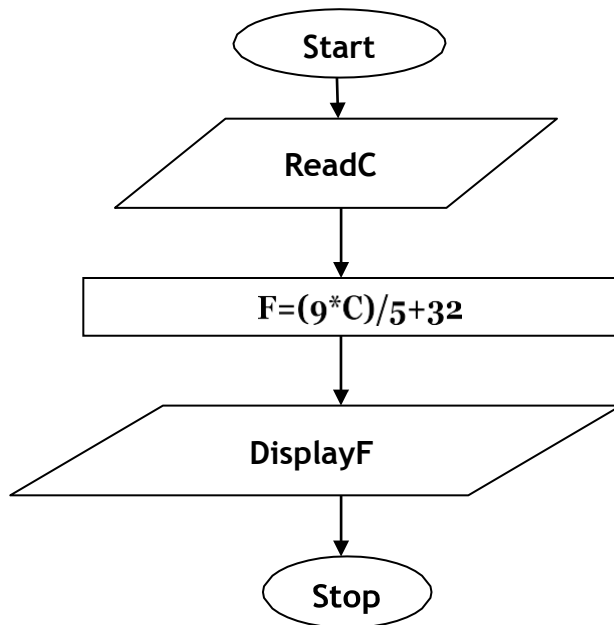


b. To convert temperature from degree Celsius to Fahrenheit. Algorithm

- Step 1: Start
- Step 2 : Read the input of temperature in Celsius (say C)
- Step 3:  $F = (9 * C) / 5 + 32$
- Step 4 : Print temperature in Fahrenheit is F
- Step 5: Stop

Flowchart

EXERCISE



ShortAnswerTypeQuestions.

**Q1** What is a flowchart? (2014-Summer)

**Q2** Give the flowchart symbols for keeping, I/O statement and decision statement.

### Long Answer Type Questions

**Q1** Draw a flowchart to find the sum of 10 random numbers. (2017-Winter)

**Q2** Write an algorithm and flowchart of the followings:

- a. To calculate square of numbers.
- b. To identify whether entered number is positive (+) or negative (-).
- c. To generate natural numbers which are divisible by 5 in between 1 and 200?
- d. To find sum of digits of a three-digit number (say 364).

\*\*\*\*\*

## CHAPTER-6: OVERVIEW OF C PROGRAMMING LANGUAGE

### Constants, Variables and Data types in C

#### The C Language

C is a professional programmer's language. It was designed to get in one's way as little as possible. Kernighan and Ritchie wrote the original language definition in their book, *The C Programming Language* (below), as part of their research at AT&T. Unix and C++ emerged from the same labs. For several years I used AT&T as my long distance carrier in appreciation of all that CS research, but hearing "thank you for using AT&T" for the millionth time has used up that goodwill.

#### Important Points

- The C Language is developed by Dennis Ritchie for creating system applications that directly interact with the hardware devices such as drivers, kernels, etc.
- C programming is considered as the base for other programming languages, that is why it is known as mother language.
- It can be defined by the following ways:
  1. Mother language
  2. System programming language
  3. Procedure-oriented programming language
  4. Structured programming language
  5. Mid-level programming language

#### 1) C as a mother language

C language is considered as the mother language of all the modern programming languages because most of the compilers, JVMs, Kernels, etc. are written in C language, and most of the programming languages follow C syntax, for example, C++, Java, C#, etc. It provides the core concepts like the array, strings, functions, file handling, etc. that are being used in many languages like C++, Java, C#, etc.

#### 2) C as a system programming language

A system programming language is used to create system software. C language is a system programming language because it can be used to do low-level programming (for example driver and kernel). It is generally used to create hardware devices, OS, drivers, kernels, etc. For example, Linux kernel is written in C.

It can't be used for internet programming like Java, .Net, PHP, etc.

#### 3) C as a procedural language

A procedure is known as a function, method, routine, subroutine, etc. A procedural language specifies a series of steps for the program to solve the problem.

A procedural language breaks the program into functions, data structures, etc.

C is a procedural language. In C, variables and function prototypes must be declared before being used.

#### 4) **C as a structured programming language**

A structured programming language is a subset of the procedural language. Structure means to break a program into parts or blocks so that it may be easy to understand.

In the C language, we break the program into parts using functions. It makes the program easier to understand and modify.

#### 5) **C as a mid-level programming language**

C is considered as a middle-level language because it supports the feature of both low-level and high-level languages. C language program is converted into assembly code, it supports pointer arithmetic (low-level), but it is machine independent (a feature of high-level).

A Low-level language is specific to one machine, i.e., machine dependent. It is machine dependent, fast to run. But it is not easy to understand.

A High-Level language is not specific to one machine, i.e., machine independent. It is easy to understand.

#### **Quick History of C**

- Developed at Bell Laboratories in the early seventies by Dennis Ritchie.
- Born out of two other languages—BCPL (Basic Control Programming Language) and B.
- C introduced such things as character types, floating point arithmetic, structures, unions and the preprocessor.
- The principal objective was to devise a language that was easy enough to understand to be "high-level"—i.e. understood by general programmers, but low-level enough to be applicable to the writing of systems-level software.
- The language should abstract the details of how the computer achieves its tasks in such a way as to ensure that C could be portable across different types of computers, thus allowing the UNIX operating system to be compiled on other computers with a minimum of re-writing.
- C as a language was in use by 1973, although extra functionality, such as new types, was introduced up until 1980.
- In 1978, Brian Kernighan and Dennis M. Ritchie wrote the seminal work *The C Programming Language*, which is now the standard reference book for C.
- A formal ANSI standard for C was produced in 1989.
- In 1986, a descendant of C, called C++ was developed by Bjarne Stroustrup, which is in wide use today. Many modern languages such as C#, Java and Perl are based on C and C++.
- Using C language scientific, business and system-level applications can be developed easily.

#### **Constants**

In C programming language, a constant is similar to the variable but the constant holds only one value during the program execution. That means, once a value is assigned to the constant, that value can't be changed during the program execution. Once the value is assigned to the constant, it is fixed throughout the program. A constant can be defined as follows...

**A constant is a named memory location which holds only one value throughout the program execution.**

In C programming language, a constant can be of any data type like integer, floating-point, character, string and double, etc.,

### Integer constants

An integer constant can be a decimal integer or octal integer or hexadecimal integer. A decimal integer value is specified as a direct integer value whereas an octal integer value is prefixed with 'O' and a hexadecimal value is prefixed with 'OX'.

An integer constant can also be an unsigned type of integer constant or long type of integer constant. An unsigned integer constant value is suffixed with 'u' and a long integer constant value is suffixed with 'l' whereas an unsigned long integer constant value is suffixed with 'ul'.

#### **Example**

125 -----> Decimal Integer Constant  
076----- > Octal Integer Constant  
OX3A----- > Hexa Decimal Integer Constant  
50u -----> Unsigned Integer Constant  
30l -----> Long Integer Constant  
100ul ----- > Unsigned Long Integer Constant

### Floating Point constants

A floating-point constant must contain both integer and decimal parts. Sometimes it may also contain the exponent part. When a floating-point constant is represented in exponent form, the value must be suffixed with 'e' or 'E'.

#### **Example**

The floating-point value 3.14 is represented as 3E-14 in exponent form.

### Character Constants

A character constant is a symbol enclosed in single quotation. A character constant has a maximum length of one character.

#### **Example**

'A'

'2'

'+'

In the C programming language, there are some predefined character constants called escape sequences. Every escape sequence has its own special functionality and every escape sequence is prefixed with '\'. These escape sequences are used in an output function called 'printf()'.

### String Constants

➤ A string constant is a collection of characters, digits, special symbols and escape sequences that are enclosed in double quotations.

➤ We define a string constant in a single line as follows. .... "This is Diploma smartclass"

➤ We can define a string constant using multiple lines as follows... "This\  
is\  
Diploma smartclass"

➤ We can also define a string constant by separating it with white spaces

follows... "This" is "Diplomasmartclass"

All the above three defines the same string constant.

### Creating constants in C

- In a programming language, constants can be created using two concepts...
  - Using the `const` keyword
  - Using `#define` preprocessor

#### Using the "const" keyword

• We create a constant of any data type using 'const' keyword. To create a constant, we prefix the variable declaration with 'const' keyword.

• The general syntax for creating constant using 'const' keyword is as follows... `const data type constantName;`

OR

`const data type constantName = value;`

#### Example

```
const int x = 10;
```

Here, 'x' is an integer constant with fixed value 10.

#### Example Program

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i = 9;
    const int x = 10; i = 15;
    x = 100; // creates an error
    printf("i = %d\n x = %d", i, x);
}
```

The above program gives an error because we are trying to change the constant variable value (`x = 100`).

#### Using '#define' preprocessor

We can also create constants using '#define' preprocessor directive. When we create constant using this preprocessor directive it must be defined at the beginning of the program (because all the preprocessor directives must be written before the global declaration).

We use the following syntax to create constant using '#define' preprocessor directive... `#define CONSTANTNAME value`

#### Example

```
#define PI 3.14
```

Here, PI is a constant with value 3.14

#### Example Program

```
#include <stdio.h>
#include <conio.h>
#define PI 3.14
void main()
{
    int r, area;
    printf("Please enter the radius of circle : ");
    scanf("%d", &r);
    area = PI * (r * r);
    printf("Area of the circle = %d", area);
}
```

}

## Variables:

Variables in a C programming language are the named memory locations where the user can store different values of the same datatype during the program execution. That means a variable is a name given to a memory location in which we can store different values of the same data type. In other words, a variable can be defined as a storage container to hold values of the same datatype during the program execution. The formal definition of a datatype is as follows.

**Variable is a name given to a memory location where we can store different values of the same datatype during the program execution.**

Every variable in C programming language must be declared in the declaration section before it is used. Every variable must have a data type that determines the range and type of values to be stored and the size of the memory to be allocated.

A variable name may contain letters, digits and underscore symbol. The following are the rules to specify a variable name...

- Variable name should not start with a digit.
- Keywords should not be used as variable names.
- A variable name should not contain any special symbols except underscore (\_).
- A variable name can be of any length but the compiler considers only the first 31 characters of the variable name.

## Declaration of Variable

Declaration of a variable tells the compiler to allocate the required amount of memory with the specified variable name and allows only specified datatype values into that memory location. In C programming language, the declaration can be performed either before the function as global variables or inside any block or function. But it must be at the beginning of a block or function.

### Declaration Syntax:

```
datatype variableName;
```

### Example

```
int number;
```

The above declaration tells to the compiler that allocates 2 bytes of memory with the name `number` and allows only integer values into that memory location.

## Datatypes

Data used in a C program is classified into different types based on its properties. In the C programming language, a datatype can be defined as a set of values with similar characteristics. All the values in a datatype have the same properties.

Data types in the C programming language are used to specify what kind of value can be stored in a variable. The memory size and type of the value of a variable are determined by the variable's datatype. In a program, each variable or constant or array must have a datatype and this datatype specifies how much memory is to be allocated and what type of values are to be stored in that variable or constant or array. The formal definition of a data type is as follows...

The datatype is a set of values with predefined characteristics. Datatypes are used to declare variables, constants, arrays, pointers, and functions.

In the C programming language, datatypes are classified as follows...

- Primary data types (Basic data types OR Predefined datatypes)
- Derived datatypes (Secondary datatypes OR User-defined datatypes)
- Enumeration datatypes



## ➤ Void datatype

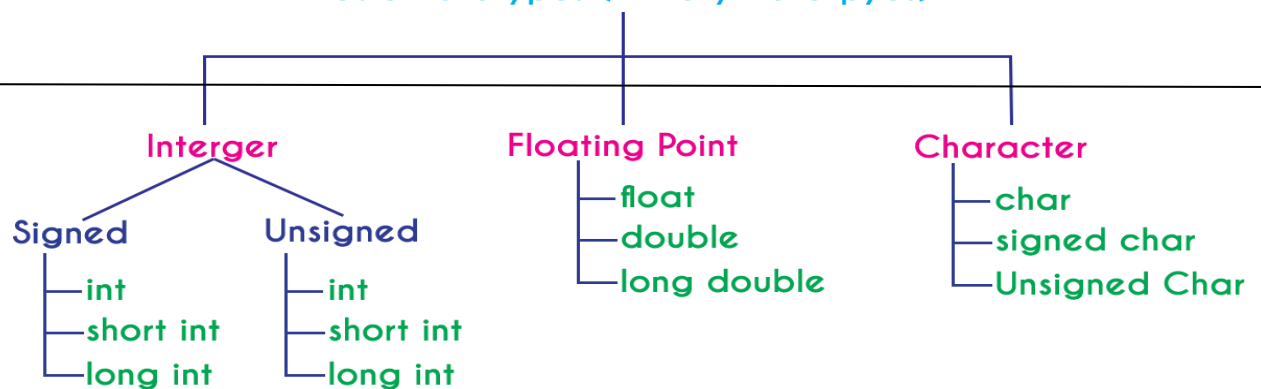
	Type	Size (Bytes)	Range	Specifier
<b>Dataty</b>	int (signed short int)	2	-32768 to +32767	%d
	short int (signed short int)	2	-32768 to +32767	%d
	long int (signed long int)	4	-2,147,483,648 to +2,147,483,647	%d
	unsigned int (unsigned short int)	2	0 to 65535	%u
	unsigned long int	4	0 to 4,294,967,295	%u

## Primary data types

The primary data types in the C programming language are the basic data types. All the primary data types are already defined in the system. Primary data types are also called as Built-In data types. The following are the primary data types in C programming language.

- Integer datatype
- Floating Point datatype

### Basic Datatypes (Primary Datatypes)



- Double datatype
- Character datatype

## Integer Datatype (int)

The integer data type is a set of whole numbers. Every integer value does not have the decimal value. We use the keyword "int" to represent integer data type in C. We use the keyword int to declare the variables and to specify the return type of a function. The integer data type is used with different type modifiers like short, long, signed and unsigned. The following table provides complete details about the integer datatype.

## Floating Point data types

Floating-point data types are a set of numbers with the decimal value. Every floating-point value must contain the decimal value. The floating-point data type has two variants...

- float

- double

We use the keyword "float" to represent floating-point data type and "double" to represent double data type in c. Both float and double are similar but they differ in the number of decimal places. The float value contains 6 decimal places whereas double value contains 15 or 19 decimal places. The following table provides complete details about floating-point data type.

### Character data type

The character data type is a set of characters enclosed in single quotations. The following table provides complete details about the character data type.

Type	Size (Bytes)	Range	Specifier
char (signed char)	1	-128 to +127	%c
unsigned char	1	0 to 255	%c

The following table provides complete information about all the data types in c programming language...

	Integer	Floating Point	Double	Character
What is it?	Numbers without decimal value	Numbers with decimal value	Numbers with decimal value	Any symbol enclosed in single quotation
Keyword	int	float	double	char
Memory Size	2 or 4 Bytes	4 Bytes	8 or 10 Bytes	1 Byte
Range	-32768 to +32767 (or) 0 to 65535 (In case of 2 bytes only)	1.2E - 38 to 3.4E + 38	2.3E-308 to 1.7E+308	-128 to +127 (or) 0 to 255
Type Specifier	%d or %i or %u	%f	%ld	%c or %s
Type Modifier	short, long signed, unsigned	No modifiers	long	signed, unsigned
Type Qualifier	const, volatile	const, volatile	const, volatil	const, volatile

### Void data type

The void data type means nothing or no value. Generally, the void is used to specify a function which does not return any value. We also use the void data type to specify empty parameters of a function.

Type	Size (Bytes)	Range	Specifier
float	4	1.2E - 38 to 3.4E + 38	%f
double	8	2.3E-308 to 1.7E+308	%ld
long double	10	3.4E-4932 to 1.1E+4932	%ld

## Enumerated datatype

An enumerated data type is a user-defined data type that consists of integer constants and each integer constant is given a name. The keyword "enum" is used to define the enumerated data type.

## Derived datatypes

Derived datatypes are user-defined datatypes. The derived datatypes are also called as user-defined data types or secondary data types. In the C programming language, the derived datatypes are created using the following concepts...

- Arrays
- Structures
- Unions

## Managing Input and Output operations.

### Output Functions

C programming language provides built-in functions to perform output operation. The output operations are used to display data on user screen (output screen) or printer or any file. The C programming language provides the following built-in output functions...

- printf()
- putchar()
- puts()
- fprintf()

### printf() function

The printf() function is used to print string or data values or a combination of string and data values on the output screen (User screen). The printf() function is a built-in function defined in a header file called "stdio.h". When we want to use printf() function in our program, we need to include the respective header file (stdio.h) using the #include statement. The printf() function has the following syntax...

#### Syntax:

```
printf("message to be display!!!");
```

**Example Program**

```
#include <stdio.h>
#include <conio.h>
void main()
```

```
{
printf("Hello! Welcome to Diploma smart class!!!");
}
```

In the above example program, we used the printf() function to print a string on the output screen.

Output of the program is **Hello! Welcome to Diploma smart class!!!**

The printf() function is also used to display data values. When we want to display data values, we use a format string of the data value to be displayed.

#### Syntax:

```
printf("format string", variableName);
```

#### Example Program:

```
#include <stdio.h>
#include <conio.h>
void main()
{
inti=10; float x
```

```
=5.5;printf("%d%f",i,x);
}
```

In the above example program, we used the printf() function to print data values of variables i and x on to the output screen. Here i is an integer variable so we have used format string %d and x is a float variable so we have used format string %f.

The printf() function can also be used to display string along with data values.

To display the output in different lines or as we wish, we use some special characters called escape sequences. Escape sequences are special characters with special functionality used in printf() function to format the output according to the user requirement. In the C programming language, we have the following escape sequences...

Escape sequence	Meaning
\n	Moves the cursor to New Line
\t	Inserts Horizontal Tab (5 characters space)
\v	Inserts Vertical Tab (5 lines space)
\a	Beep sound
\b	Backspace (removes the previous character from its current position)
\\	Inserts Backward slash symbol
\?	Inserts Question mark symbol
\'	Inserts Single quotation mark symbol
\"	Inserts Double quotation mark symbol

Consider the following example program... **Example Program:**

**Program:**

```
#include<stdio.h>#include<
conio.h>voidmain()
{
printf("Welcome
to\n");printf("Diplomasmartclass\n");
printf("thepperfectwebsiteforlearning");
}
```

**Output:-**

```
WelcometoDiplomasmartclas
s
thepperfect website forlearning
```

**putchar() function**

The putchar() function is used to display a single character on the output screen. The putchar() function prints the character which is passed as a parameter to it and returns the same character as a return value. This function is used to print only a single character. To print multiple characters we need to write multiple times or use a looping statement. Consider the following example program...

```
#include<stdio.h>#include<
conio.h>voidmain()
{
char ch =
```

```
'A';putchar(ch);
}
```

### **puts()function**

The puts() function is used to display a string on the output screen. The puts() function prints a string or sequence of characters till the newline. Consider the following example program...

#### **ExampleProgram**

```
#include<stdio.h>#include<
conio.h>voidmain()
{
char name[30];
printf("\nEnter your favourite website:
");gets(name);
puts(name);
}
```

### **fprintf()function**

The fprintf() function is used with the concept of files. The fprintf() function is used to print a line into the file. When you want to use fprintf() function the file must be opened in writing mode.

## **InputFunctions**

C programming language provides built-in functions to perform input operations. The input operations are used to read user values (input) from the keyboard. The C programming language provides the following built-in input functions.

- scanf()
- getchar()
- getch()
- gets()
- fscanf()

### **scanf()function**

The scanf() function is used to read multiple data values of different data types from the keyboard. The scanf() function is a built-in function defined in a header file called "stdio.h". When we want to use scanf() function in our program, we need to include the respective header file (stdio.h) using #include statement. The scanf() function has the following syntax...

#### **Syntax:**

```
scanf("formatstrings",&variableNames);
```

#### **ExampleProgram**

```
#include<stdio.h>#include<c
onio.h>voidmain()
{
inti;
printf("\nEnter any integer value:
");scanf("%d",&i);
printf("\nYou have entered %d number",i);
}
```

In the above example program, we used the scanf() function to read an integer value from the keyboard and store it into variable 'i'.

#### **Output:-**

**Enter any integer value:**

**53 You have entered 53 number**

The scanf() function is also used to read multiple data values of different or the same data types. Consider the following example program...

### ExampleProgram

```
#include<stdio.h>#include<conio.h>voidmain()
{
int i;float x;
printf("\nEnter one integer followed by one float value : ");scanf("%d%f",&i,&x);
printf("\nintegervalue=%d,floatvalue=%f",i, x);
}
```

#### Output:-

Enter one integer followed by one float value : 59

32.8integervalue=59,floatvalue=32.8

Intheaboveexampleprogram,weusedthescanf()functiontoreadoneintegervalueand one float value from the keyboard. Here 'i' is an integer variable so we have used formatstring %d,and'x' isa floatvariablesowehaveusedformatstring%f.

The scanf() function returns an integer value equal to the total number of input valuesreadusingscanffunction.

```
ExampleProgram#inclu
de<stdio.h>#include<con
io.h>voidmain()
{
int
i,a,b;floatx;
printf("\nEnter two integers
andonefloat:");i=scanf("%d%d%f",&a,&b,&x);printf(
"\nTotalinputs read:%d",i);
}
```

#### getchar()function:

The getchar() function is used to read a character from the keyboard and return it to the program. This function is used to read a single character. To read multiple characters we needtowritemultipletimesor use alooping statement. Consider thefollowing exampleprogram...

#### ExampleProgram

```
#include<stdio.h>#includ
e<conio.h>voidmain()
{
charch;
printf("\nEnter any character :
");ch=getchar();
printf("\nYou haveentered :%c\n",ch);
}
```

#### Output:-

Enter any character

:HYouhaveentered:H

#### getch()function

The getch() function is similar to getchar function. The getch() function is used to read a character from the keyboard and return it to the program. This function is used to read a singlecharacter.Toreadmultiplecharactersweneedtowritemultipletimesorusealoopingstatement.Considerthe followingexampleprogram...

### ExampleProgram

```
#include<stdio.h>#include
<conio.h>voidmain()
{
charch;
printf("\nEnter anycharacter:");

ch=getch();
printf("\nYou haveentered :%c",ch);
}
```

### gets()function

The gets() function is used to read a line of string and stores it into a character array. Thegets() function reads a line of string or sequence of characters till a newline symbol enters.Considerthefollowingexampleprogram...

### ExampleProgram

```
#include<stdio.h>#include
<conio.h>voidmain()
{
char name[30];
printf("\nEnter your favourite website:
");gets(name);
printf("%s",name);
}
```

### fscanf()function

Thefscanf()functionisusedwiththeconceptoffiles.Thefscanf()functionisusedtoread data values from a file. When you want to use fscanf() function the file must be opened inreadingmode.

## Operators,Expressions,Typeconversion&Typecasting

### Operators

An operator is a symbol used to perform arithmetic and logical operations in a program.That means an operator is a special symbol that tells the compiler to perform mathematical orlogical operations. C programming language supports a rich set of operators that are classifiedas follows.

- ArithmeticOperators
- RelationalOperators
- LogicalOperators
- Increment&DecrementOperators
- AssignmentOperators
- BitwiseOperators
- Conditional Operator
- SpecialOperators

### ArithmeticOperators(+, -,\*,/,%)

The arithmetic operators are the symbols that are used to perform basic mathematicaloperationslikeaddition,subtraction,multiplication,divisionandpercentagemodulo.Thefollowi ngtableprovidesinformationabout arithmeticoperators.

Operator	Meaning	Example
+	Addition	10+5= 15

-	Subtraction	10-5= 5
*	Multiplication	10*5=50
/	Division	10/5=2
%	Remainder of the Division	5%2= 1

□ The addition operator can be used with numerical data types and character data type. When it is used with numerical values, it performs mathematical addition and when it is used with character data type values, it performs concatenation (appending).

□ The remainder of the division operator is used with integer data type only.

### Relational Operators (<, >, <=, >=, ==, !=)

The relational operators are the symbols that are used to compare two values. That means the relational operators are used to check the relationship between two values. Every relational operator has two results TRUE or FALSE. In simple words, the relational operators are used to define conditions in a program. The following table provides information about relational operators.

Operator	Meaning	Example
<	Returns TRUE if the first value is smaller than second value otherwise returns FALSE	10 < 5 is FALSE
>	Returns TRUE if the first value is larger than second value otherwise returns FALSE	10 > 5 is TRUE
<=	Returns TRUE if the first value is smaller than or equal to second value otherwise returns FALSE	10 <= 5 is FALSE
>=	Returns TRUE if the first value is larger than or equal to second value otherwise returns FALSE	10 >= 5 is TRUE
==	Returns TRUE if both values are equal otherwise returns FALSE	10 == 5 is FALSE
!=	Returns TRUE if both values are not equal otherwise returns FALSE	10 != 5 is TRUE

### Logical Operators (&&, ||, !)

The logical operators are the symbols that are used to combine multiple conditions into one condition. The following table provides information about logical operators.

Operator	Meaning	Example
----------	---------	---------



&&	Logical AND- Returns TRUE if all conditions are TRUE otherwise returns FALSE	10<5&&12>10 is FALSE
	Logical OR- Returns FALSE if all conditions are FALSE otherwise returns TRUE	10<5  12>10 is TRUE
!	Logical NOT- Returns TRUE if condition is FALSE and returns FALSE if it is TRUE	!(10<5&&12>10) is TRUE

□ Logical AND- Returns TRUE only if all conditions are TRUE, if any of the conditions is FALSE then the complete condition becomes FALSE.

□ Logical OR- Returns FALSE only if all conditions are FALSE, if any of the conditions is TRUE then the complete condition becomes TRUE.

### Increment & Decrement Operators (++ & --)

The increment and decrement operators are called unary operators because both need only one operand. The increment operators add one to the existing value of the operand and the decrement operator subtracts one from the existing value of the operand. The following table provides information about increment and decrement operators.

Operator	Meaning	Example	
++	Increment	Adds one to existing value	inta=5;a++;⇒ a=6
--	Decrement	Subtracts one from existing value	inta=5;a--;⇒a=4

The increment and decrement operators are used in front of the operand (++a) or after the operand (a++). If it is used in front of the operand, we call it as pre-increment or pre-decrement and if it is used after the operand, we call it as post-increment or post-decrement.

### Pre-Increment or Pre-Decrement

In the case of pre-increment, the value of the variable is increased by one before the expression evaluation. In the case of pre-decrement, the value of the variable is decreased by one before the expression evaluation. That means, when we use pre-increment or pre-decrement, first the value of the variable is incremented or decremented by one, then the modified value is used in the expression evaluation.

### Example Program

```
#include<stdio.h>#include
<conio.h>voidmain()
{
inti=5,j;
j=++i; //Pre-
```

```
Incrementprintf("i=%d,j=%d",i,j);
}
```

### Post-IncrementorPost-Decrement

In the case of post-increment, the value of the variable is increased by one after the expression evaluation. In the case of post-decrement, the value of the variable is decreased by one after the expression evaluation. That means, when we use post-increment or post-decrement, first the expression is evaluated with existing value, then the value of the variable is incremented or decremented by one.

### Example Program

```
#include<stdio.h>#include
<conio.h>voidmain()
{
inti=5,j;
j=i++;//Post-
Incrementprintf("i=%d,j=%d",i,j);
}
```

### Assignment Operators(=,+ =,- =,\* =,/ =,% =)

The assignment operators are used to assign right-hand side value (Rvalue) to the left-hand side variable (Lvalue). The assignment operator is used in different variants along with arithmetic operators. The following table describes all the assignment operators in the C programming language.

Operator	Meaning	Example
=	Assign the right-hand side value to left-hand side variable	A=15
+ =	Add both left and right-hand side values and store the result into left-hand side variable	A+=10⇒A=A+10
- =	Subtract right-hand side value from left-hand side variable value and store the result into left-hand side variable	A-=B⇒A= A-B
* =	Multiply right-hand side value with left-hand side variable value and store the result into left-hand side variable	A*=B⇒A=A*B
/ =	Divide left-hand side variable value with right-hand side variable value and store the result into the left-hand side variable	A/=B⇒A=A/B
% =	Divide left-hand side variable value with right-hand side variable value and store the remainder into the left-hand side variable	A%=B⇒A=A%B

## Bitwise Operators(&,|,^,~,>>, <<)

The bitwise operators are used to perform bit-level operations in the C programming language. When we use the bitwise operators, the operations are performed based on the binary values. The following table describes all the bitwise operators in the C programming language.

Let us consider two variables A and B as A= 25(11001) and B=20 (10100).

Operator	Meaning	Example
&	the result of Bitwise AND is 1 if all the bits are 1 otherwise it is 0	A&B ⇒ 16(10000)
	the result of Bitwise OR is 0 if all the bits are 0 otherwise it is 1	A B ⇒ 29(11101)
^	the result of Bitwise XOR is 0 if all the bits are same otherwise it is 1	A^B ⇒ 13(01101)
~	the result of Bitwise one's complement is negation of the bit (Flipping)	~A ⇒ 6(00110)
<<	the Bitwise left shift operator shifts all the bits to the left by the specified number of positions	A<<2 ⇒ 100(100100)
>>	the Bitwise right shift operator shifts all the bits to the right by the specified number of positions	A>>2 ⇒ 6(00110)

## Conditional Operator(?:)

The conditional operator is also called a ternary operator because it requires three operands. This operator is used for decision making. In this operator, first we verify a condition, then we perform one operation out of the two operations based on the condition result. If the condition is TRUE the first option is performed, if the condition is FALSE the second option is performed. The conditional operator is used with the following syntax.

**Condition?TRUE Part:FALSE Part;**

### Example

A = (10 < 15) ? 100 : 200; ⇒ A value is 100

## Special Operators(sizeof, pointer, comma, dot, etc.)

The following are the special operators in C programming language.

### sizeof operator

This operator is used to find the size of the memory (in bytes) allocated for a variable.

This operator is used with the following syntax: `sizeof(variableName);`

### **Example**

`sizeof(A);` ⇒ the result is 2 if A is an integer

### **Pointer operator (\*)**

This operator is used to define pointer variables in C programming language.

### **Comma operator (,)**

This operator is used to separate variables while they are declaring, separate the expressions in function calls, etc.

### **Dot operator (.)**

This operator is used to access members of structure or union.

## **Operator Precedence and Associativity**

### **What is Operator Precedence?**

Operator precedence is used to determine the order of operators evaluated in an expression. In programming language every operator has precedence (priority). When there is more than one operator in an expression the operator with higher precedence is evaluated first and the operator with the least precedence is evaluated last.

### **What is Operator Associativity?**

Operator associativity is used to determine the order of operators with equal precedence evaluated in an expression. In the C programming language, when an expression contains multiple operators with the equal precedence, we use associativity to determine the order of evaluation of those operators.

In programming language the operator precedence and associativity are shown in the following table.

Precedence	Operator	Operator Meaning	Associativity
1	() [] -> .	function call array reference structure member access structure member access	Left to Right
2	! ~ + - ++ -- & * sizeof(type)	negation 1's complement Unary plus Unary minus increment operator decrement operator address of operator pointer return size of variable type conversion	Right to Left
3	* / %	multiplication division remainder	Left to Right
4	+ -	addition subtraction	Left to Right
5	<< >>	left shift right shift	Left to Right
6	< <= > >=	less than less than or equal to greater than greater than or equal to	Left to Right
7	== !=	equal to not equal to	Left to Right
8	&	bitwise AND	Left to Right
9	^	bitwise EXCLUSIVE OR	Left to Right
10		bitwise OR	Left to Right
11	&&	logical AND	Left to Right
12		logical OR	Left to Right
13	?:	conditional operator	Left to Right
14	= *= /= %= += -= &= ^=  = <<= >>=	assignment assignment multiplication assignment division assign remainder assignment addition assignment subtraction assignment bitwise AND assignment bi twise XOR assignment bitwise OR assignment lefts hift assignment right shift	Right to Left
15	,	separator	Left to Right

In the above table, the operator precedence decreases from top to bottom and increases from bottom to top.

## Expressions

### What is an expression?

In any programming language, if we want to perform any calculation or to frame any condition etc., we use a set of symbols to perform the task. These set of symbols makes an expression.

In the C programming language, an expression is defined as follows.

**An expression is a collection of operators and operands that represents a specific value.**

In the above definition, an operator is a symbol that performs tasks like arithmetic operations, logical operations, and conditional operations, etc.

Operands are the values on which the operators perform the task. Here operand can be a direct value or variable or address of memory location.

### Expression Types in C

In the C programming language, expressions are divided into THREE types. They are as follows...

- Infix Expression
- Postfix Expression
- Prefix Expression

The above classification is based on the operator position in the expression.

### Infix Expression

- The expression in which the operator is used between operands is called infix expression.
- The infix expression has the following general structure.

### Operand1 Operator Operand2

### Postfix Expression

- The expression in which the operator is used after operands is called postfix expression.
- The postfix expression has the following general structure.

### Operand1 Operand2 Operator

### Prefix Expression

- The expression in which the operator is used before operands is called a prefix expression.
- The prefix expression has the following general structure.

### Operator Operand1 Operand2

## Type Conversion and Type Casting

In a programming language, the expression contains data values of the same data type or different data types. When the expression contains similar data type values then it is evaluated without any problem. But if the expression contains two or more different data type values then they must be converted to the single data type of destination data type. Here, the destination is the location where the final result of that expression is stored. For example, the multiplication of an integer data value with the float data value and storing the result into a float variable. In this case, the integer value must be converted to float value so that the final result is a float data type value.

In a programming language, the data conversion is performed in two different methods

as follows...

- Type Conversion
- Type Casting

### Type Conversion

The type conversion is the process of converting a data value from one data type to another data type automatically by the compiler. Sometimes type conversion is also called implicit type conversion. The implicit type conversion is automatically performed by the compiler.

For example, in C programming language, when we assign an integer value to a float variable the integer value automatically gets converted to float value by adding decimal value 0. And when a float value is assigned to an integer variable the float value automatically gets converted to an integer value by removing the decimal value. To understand more about type conversion observe the following...

```
int i=10; float x = 15.5
; char ch='A' ;
i = x ; =====> x value 15.5 is converted as 15 and assigned to variable i
x = i; =====> Here i value 10 is converted as 10.000000 and assigned to variable x
i = ch; =====> Here the ASCII value of A(65) is assigned to i
```

### Example Program

```
#include <stdio.h> #include <conio.h>
void main()
{
    int i=95 ;
    float x = 90.99
    ; char ch='A' ;
    i=x;
    printf("i value is %d\n",i); x=i;
    printf("x value is %f\n",x); i=ch;
    printf("i value is %d\n",i);
}
```

In the above program, we assign  $i = x$ , i.e., float variable value is assigned to the integer variable. Here, the compiler automatically converts the float value (90.99) into integer value (90) by removing the decimal part of the float value (90.99) and then it is assigned to variable  $i$ . Similarly, when we assign  $x = i$ , the integer value (90) gets converted to float value (90.000000) by adding zero as the decimal part.

### Typecasting

Typecasting is also called an explicit type conversion. Compiler converts data from one data type to another data type implicitly. When compiler converts implicitly, there may be a data loss. In such a case, we convert the data from one data type to another data type using explicit type conversion. To perform this we use the unary cast operator. To convert data from one type to another type we specify the target data type in parenthesis as a prefix to the data value that has to be converted.

The general syntax of typecasting is as follows.

**(Target Datatype) Data Value**

### Example

```
int totalMarks = 450, maxMarks = 600; float average;
average = (float) totalMarks / maxMarks * 100;
```

In the above example code, both totalMarks and maxMarks are integer data values. When we perform  $\text{totalMarks} / \text{maxMarks}$  the result is a float value, but the destination (average) data type is a float. So we use type casting to convert totalMarks and maxMarks into float data type.

### Example Program

```
#include <stdio.h> #include <conio.h>
int main()
{
    int a, b, c ; float avg ;
    printf( "Enter any three integer values :
```

```
");scanf(—%d%d%d%dl,a,b,c);
avg =(a+b+c)/3;
printf("avg before casting = &f", avg \n);avg
=(float)(a+ b+c) /3;
printf("avg after casting = %f",avg\n);return0;
}
```

### Comments

Comments in C are enclosed by slash/star pairs: /\* .. comments .. \*/ which may crossmultiple lines. C++ introduced a form of comment started by two slashes and extending to theend oftheline:

```
// commentuntiltheline end
```

The//commentformissohandythatmanyCcompilersnowalsosupportit,althoughitisnottechnically partoftheClanguage.

Along with well-chosen function names, comments are an important part of well writtencode. Comments should not just repeat what the code says. Comments should describe whatthecodeaccomplisheswhichmuchmoreisinterestingthanatranslationofwhateachstatementdoes.Comme ntsshouldalsonarratewhatistrickyornon-obviousaboutasectionofcode.

### DecisionControlandLoopingStatements(If,If-else,If-else-if,Switch,While,Do-while,For,Break, Continue&Goto)

#### ControlStructures

C uses curly braces ({}) to group multiple statements together. The statements execute in order. Some languages let you declare variables on any line (C++). Other languages insist thatvariables are declared only at the beginning of functions (Pascal). C takes the middle road --variables may be declaredwithin the body ofa function, but they must follow a '{'. Moremodern languages like Java and C++ allow you to declare variables on any line, which is handy.

#### WhatisDecisionMakingStatement?

In the C programming language, the program execution flow is line by line from top tobottom. That means the c program is executed line by line from the main method. But this typeof execution flow may not be suitable for all the program solutions. Sometimes, we make somedecisions or we may skip the execution of one or more lines of code. Consider a situation,where we write a program to check whether a student has passed or failed in a particularsubject. Here, we need to check whether the marks are greater than the pass marks or not. Ifmarks are greater, then we decide that the student has passed otherwise failed. To solve suchkindofproblemsincweuse thestatementscalleddecision makingstatements.

**Decision-making statements are the statements that are used to verify a given conditionand decide whether a block of statements gets executed or not based on the conditionresult.**

Inthecprogramming language,therearetwodecision-makingstatementstheyare asfollows.

1. ifstatement
2. switch statement

#### ifstatementinc

Inc,ifstatementisusedtomakedecisionsbasedonacondition.Theifstatementverifiesthegivenconditionanddec ideswhetherablockofstatementsareexecutedornotbasedontheconditionresult.Inc,ifstatement isclassifiedinto fourtypesasfollows...

1. Simpleifstatement
2. if-elsestatement
3. Nested ifstatement
4. if-else-ifstatement(if-elseladder)

#### Simpleif statement

Simpleifstatementisusedtoverifythegivenconditionandexecutestheblockofstatements based on the condition result. The simple if statement evaluates specified condition.If it is TRUE, it executes the next



statement or block of statements. If the condition is FALSE, it skips the execution of the next statement or block of statements. The general syntax and execution flow of the simple if statement is as follows.

Simple if statement is used when we have only one option that is executed or skipped based on a condition.

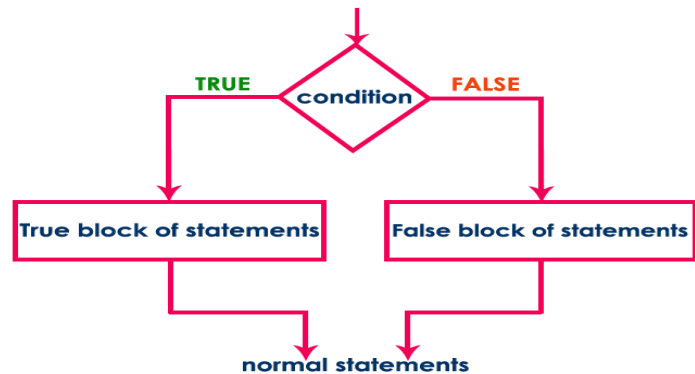
### if-else statement

The if-else statement is used to verify the given condition and executes only one out of the

#### Syntax

```
if ( condition )
{
  ....
  True block of statements;
  ....
}
else
{
  ....
  False block of statements;
  ....
}
```

#### Execution flow diagram



two blocks of statements based on the condition result. The if-else statement evaluates the specified condition. If it is TRUE, it executes a block of statements (True block). If the condition is FALSE, it executes another block of statements (False block). The general syntax and execution flow of the if-else statement is as follows.

The if-else statement is used when we have two options and only one option has to be executed based on a condition result (TRUE or FALSE).

### Nested if statement

Writing an if statement inside another if statement is called a nested if statement. The general syntax of the nested if statement is as follows.

#### Syntax

```
if ( condition1 )
{
  if ( condition2 )
  {
    ....
    True block of statements 1;
  }
  ....
}
else
{
  False block of condition1;
}
```

The nested if statement can be defined using any combination of simple if & if-else statements.

### **if-else-if statement (if-else ladder)**

Writing an if statement inside another if statement is called an if-else-if statement. The general syntax of the if-else-if statement is as follows...

#### **Syntax**

```
if ( condition1 )
{
    ....
    True block of statements1;
    ....
}
else if ( condition2 )
{
    False block of condition1;
    &
    True block of condition2
}
```

The if-else-if statement can be defined using any combination of simple if & if-else statements.

### **Examples:**

#### **Program 1: (Simple if statement)**

```
#include <stdio.h>
int main()
{
    int x=20;
    int y=22;          if (x<y)
    {
        printf("Variable x is less than y");
    }
    return 0;
}
```

#### **Output:**

**Variable x is less than y**

#### **Program 2: (Simple if statement)**

```
#include <stdio.h>
int main()
{
    int i = 10;
    if (i > 15)
    {
        printf("10 is less than 15");
    }
    printf("I am not in if");
}
```

```
}
```

**Output:** IamNotinif

### **Program3:(if-elsestatement)**

```
#include
<stdio.h>intmain()
{
int i =
20;if(i< 15)
printf("i is smaller than 15");else
printf("i is greater than 15");return0;
}
```

**Output:** iis greaterthan15

```
Program 4:(if-else
statement)#include <stdio.h>
intmain()
{
intage;
printf("Enter your
age:");scanf("%d",&age);if(a
ge>=18)
{
/* Thisstatementwillonlyexecuteif the
*abovecondition(age>=18)returnstrue
*/
printf("Youareeligibleforvoting");
}
else
{
/* Thisstatementwillonlyexecuteif the
* condition specified in the "if" returns false.
*/printf("Youarenoteligible forvoting");
}
return0;
}
```

Output:Enteryour age:14Youare noteligiblefor

### votingProgram5:(Nested-ifstatement)

```
#include
<stdio.h>intmain()
{
intvar1,var2;
printf("Input the value of
var1:");scanf("%d",&var1);printf("In
putthe valueofvar2:");
```

```
scanf("%d",&var2);if(v
ar1!=var2)
{
printf("var1 is not equal tovar2\n");

//Nested
ifelseif(var1>var2)
{
printf("var1isgreaterthanvar2\n");
}
else
{
printf("var2isgreaterthanvar1\n");
}
}
}
```

### Output:

Input the valueofvar1:12Input the  
valueofvar2:21var1  
isnotequaltovar2var2  
isgreaterthanvar1

### Program6:(Nested-ifstatement)

```
#include <stdio.h>intmain()
{
inti=10;if(i==10)
{
// First if statementif(i<15)
printf("iis smallerthan15\n");
// Nested-ifstatement will onlybeexecutedifstatementaboveis trueif(i<12)
printf("i is smaller than 12 too\n");else
printf("iis greater than15");
}
return0;
}
```

### Output:

i is smaller than 15  
i issmallerthan12 too

### Program7:(if-else-ifstatement)

```
#include <stdio.h>voidmain()
{
inti=20;if(i==10)

printf("i is 10");else if (i ==
15)printf("i is 15");else if (i
== 20)printf("i is 20");else
printf("iisnotpresent");
}
```

**Output:**

iis20

**Program8:(if-else-ifstatement)**

```
#include <stdio.h>intmain()
{
    int var1,var2;
printf("Input the value of
var1:");scanf("%d",&var1);printf("Input the
value of var2:");scanf("%d",&var2);
if(var1!=var2)
{
printf("var1 isnotequaltovar2\n");
}
else
if(var1 > var2)
{
printf("var1 isgreaterthanvar2\n");
}
else
if(var2 > var1)
{
printf("var2 isgreaterthanvar1\n");
}
else
{
printf("var1 isequaltovar2\n");
}
return0;
}
```

**Output:**

Input the value of var1:12Input the value  
of var2:21var1 isnotequaltovar2

**'switch'statement inC**

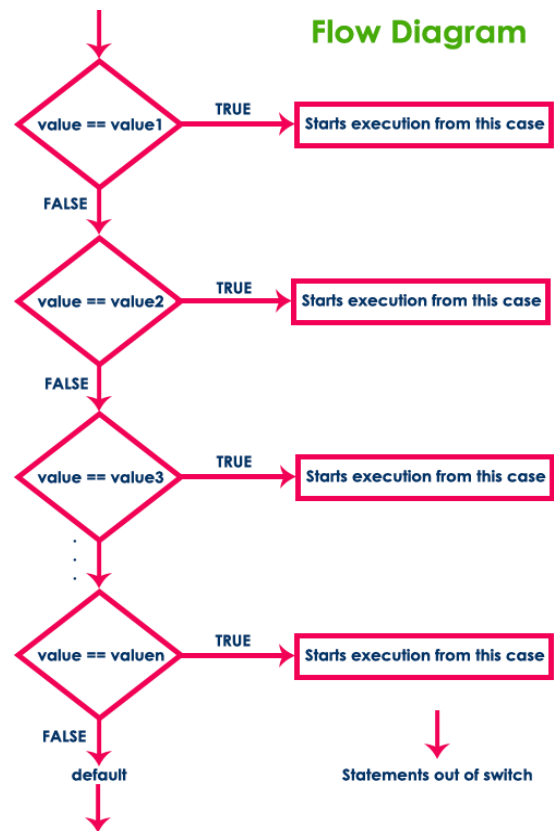
Consider a situation in which we have many options out of which we need to select only one option that is to be executed. Such kind of problems can be solved using nested if statement. But as the number of options increases, the complexity of the program also gets increased. This type of problem can be solved very easily using a switch statement. Using the switch statement, one can select only one option from more number of options very easily. In the switch statement, we provide a value that is to be compared with a value associated with each option. Whenever the given value matches the value associated with an option, the execution starts from that option. In the switch statement, every option is defined as a case.

The switch statement has the following syntax and execution flow diagram.

## Syntax

```
switch ( expression or value )
{
    case value1: set of statements;
    ....
    case value2: set of statements;
    ....
    case value3: set of statements;
    ....
    case value4: set of statements;
    ....
    case value5: set of statements;
    ....
    .
    .
    default: set of statements;
}
```

## Flow Diagram



The switch statement contains one or more cases and each case has a value associated with it. At first switch statement compares the first case value with the switchValue, if it gets matched the execution starts from the first case. If it doesn't match the switch statement compares the second case value with the switch Value and if it is matched the execution starts from the second case. This process continues until it finds a match. If no case value matches with the switchValue specified in the switch statement, then a special case called default is executed. When a case value matches with the switch Value, the execution starts from that particular case. This execution flow continues with the next case statements also. To avoid this, we use the "break" statement at the end of each case. That means the break statement is used to terminate the switch statement. However, it is optional.

## Examples:

### Program1:

```
#include <stdio.h> #include
<conio.h> void main()
{
    int n; clrscr();
    printf("Enter any digit: ")
    ; scanf("%d", &n);
    switch( n)
    {
        case 0: printf("ZERO");

        break;
        case 1: printf("ONE") ; break;
        case 2: printf("TWO");

        break;
        case 3: printf("THREE") ; break;
        case 4: printf("FOUR") ; break;
        case 5: printf("FIVE") ; break;
        case 6: printf("SIX") ; break;
```

```

case 7: printf("SEVEN") ;break;
case 8: printf("EIGHT") ;break;
case9:printf("NINE") ;
break; default: printf("NotaDigit");
}
getch() ;
}

```

### **Output:**

Enter any digit: 5FIVE

### **Program2:**

```

#include
<stdio.h>intmain()
{
intnum=2;switch(nu
m+2)
{
case 1:printf("Case1: Value is: %d", num);case
2:printf("Case2: Value is: %d", num);case
3:printf("Case3: Value is: %d",
num);default:printf("Default:Value is:%d",num);
}
return0;
}

```

Output: Default: valueis: 2

### **Loopingstatements**

Consider a situation in which we execute a single statement or block of statements repeatedly for the required number of times. Such kind of problems can be solved using looping statements in C. For example, assume a situation where we print a message 100 times. If we want to perform that task without using looping statements, we have to either write 100 printf statements or we have to write the same message 100 times in a single printf statement. Both are complex methods. The same task can be performed very easily using looping statements.

**The looping statements are used to execute a single statement or block of**

**statements repeatedly until the given condition is FALSE.**

C language provides three looping statements...

- while statement
- do-while statement
- for statement

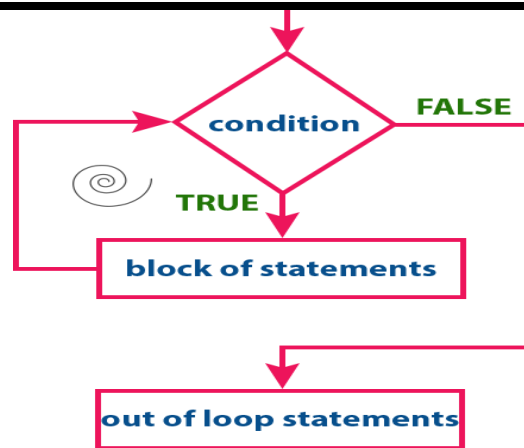
### **while Statement**

The while statement is used to execute a single statement or block of statements repeatedly as long as the given condition is TRUE. The while statement is also known as Entry control looping statement. The while statement has the following syntax... The while statement has the following execution flow diagram...

At first, the given condition is evaluated. If the condition is TRUE, the single statement or block of statements get executed. Once the execution gets completed the condition is evaluated

## Syntax:

```
while( condition )
{
    ...
    block of statements;
    ...
}
```



again. If it is TRUE, again the same statements get executed. The same process is repeated until the condition is evaluated to FALSE. Whenever the condition is evaluated to FALSE, the execution control moves out of the while block.

## Examples:

### Program1:

```
#include<stdio.h>#inclu
de<conio.h>voidmain()
{
int n = 0;clrscr();
printf("Even numbers upto 10
are");while(n<=10)
{
if(n%2==0)printf("%d\
t", n) ;n++;
}
tch() ;
}
```

### Output:

Even numbers upto 10 are 0 2 4 6 8 10

### Program2:

```
#include <stdio.h>intmain()
{
intcount=1;
while(count<= 4)
{
printf("%d ", count);count++;
}
return0;
}
```

### Output:

1234

## 'do-while' statement

The do-while statement is used to execute a single statement or block of statements repeatedly as long as

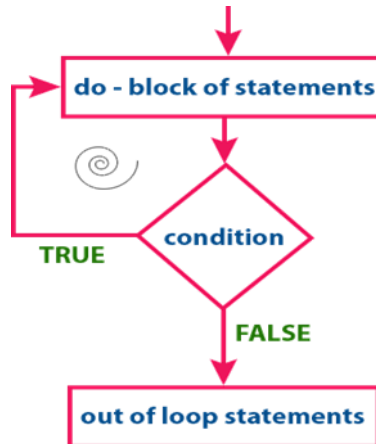


given the condition is TRUE. The do-while statement is also known as the Exit control looping statement. The do-while statement has the following syntax...

**Syntax:**

```
do
{
    ...
    block of statements;
    ...
} while( condition ) ;
```

The do-while statement has the following execution flow diagram...



At first, the single statement or block of statements which are defined in do block are executed. After the execution of the do block, the given condition gets evaluated. If the condition is evaluated to TRUE, the single statement or block of statements of do block are executed again.

Once the execution gets completed again the condition is evaluated. If it is TRUE, again the same statements are executed. This same process is repeated until the condition is evaluated to FALSE. Whenever the condition is evaluated to FALSE, the execution control moves out of the while block.

**Examples:**

**Program 1:**

```
#include <stdio.h>
int main()
{
    int j=0;
    do
    {
        printf("Value of variable j is: %d\n", j);
        j++;
    }
    while (j<=3);
    return 0;
}
```

**Output:**

```
Value of variable j is: 0
Value of variable j is: 1
Value of variable j is: 2
Value of variable j is: 3
```

**Program 2:**

```
#include <stdio.h>intmain()
{
int i=0;do
{
printf("whilevsdo-while\n");
}
while(i==1);printf("Outofloop");
}
```

**Output:**

while vs do-whileOutofloop

**'for' statement**

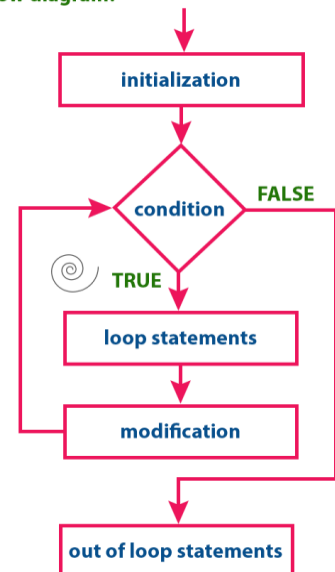
Theforstatementisusedtoexecuteasinglestatement or a block ofstatements repeatedly as longasthegivenconditionisTRUE.Theforstatementhas thefollowing syntax andexecutionflowdiagram...

Atfirst,theforstatementexecutesinitializationfollowedbycondition evaluation.IftheconditionisevaluatedtoTRUE,thesinglestatementor blockofstatementsofforstatementareexecuted.Oncetheexecutiongets completed,thomodificationstatementisexecuted and again the condition is evaluated. If it is TRUE,againthesamestatementsareexecuted.ThesameprocessisrepeateduntiltheconditionisevaluatedtoFALSE. Whenever the condition is evaluated to FALSE, theexecutioncontrolmovesoutofthefor block.

Syntax:

```
for( initialization ; condition ; modification )
{
...
block of statements;
...
}
```

Execution flow diagram:



**Examples:**

**Program1:**

```
//Printthe naturalnumbersfrom1to 10#include <stdio.h>
intmain()
{
inti;
for(i=1;i< 11;++i)
{
printf("%d",i);
}
return0;
}
```

**Output:**

12345678910

**Program2:**

```
#include <stdio.h>intmain()
{
for(int i=0; i<2; i++)
{
for(int j=0; j<4; j++)
{
printf("%d,%d\n",i,j);
}
}
return0;
}
```

```
}
```

**Output:**

0,00,10,20,31,01,11,21,3

**break,continue and goto in C**

In c, there are control statements that do not need any condition to control the program execution flow. These control statements are called as unconditional control statements. C programming language provides the following unconditional control statements...

- break
- continue
- goto

The above three statements do not need any condition to control the program execution flow.

**“break” statement**

In C, the break statement is used to perform the following two things...

- break statement is used to terminate the switch case statement
- break statement is also used to terminate looping statements like while, do-while and for.

When a break statement is encountered inside the switch case statement, the execution control moves out of the switch statement directly. For example, consider the following program.

**Examples:****Program1:**

```
#include <stdio.h> #include
<stdlib.h> void main()
{
inti;
for(i = 0; i < 10; i++)
{
printf("%d
", i); if(i == 5) break;
}
printf("\n came outside of loop i = %d", i);
}
```

**Output:**

012345
came outside of loop i = 5

**Program2:**

```
#include <stdio.h> int main()
{
/* local variable definition */ int a = 10;
/* while loop execution */ while(a < 20)
{

printf("value of a: %d\n", a); a++;
if(a > 15)
{
break;
}
```

```

/* terminate the loop using break statement */
}
}
return 0;
}

```

### Output:

```

value of a: 10value of a:
11value of a: 12value of a:
13value of a: 14value of a: 15

```

### continue statement

The continue statement is used to move the program execution control to the beginning of the looping statement. When the continue statement is encountered in a looping statement, the execution control skips the rest of the statements in the looping block and directly jumps to the beginning of the loop. The continue statement can be used with looping statements like while, do while and for.

When we use continue statement with while and do-while statements the execution control directly jumps to the condition. When we use continue statement with for statement the execution control directly jumps to the modification portion (increment/decrement/any modification) of the for loop.

### Examples: Program 1:

```

#include <stdio.h> void main()
{
    int i=0; while(i!=10)
    {
        printf("%d", i); continue;
        i++;
    }
}

```

### Output:

Infinite loop

### Program 2:

```

#include <stdio.h> int main()
{
    int i=1; //initializing local variable
    //starting loop from 1 to 10
        for(i=1; i<=10; i++)
    {
        if(i==5) //if value of i is equal to 5, it will continue the
        loop continue;
    }
    printf("%d\n", i); //end of for
}
return 0;
}

```

### Output:

1  
2  
3  
4  
6  
7  
8  
9  
10

### **goto statement**

The goto statement is used to jump from one line to another line in the program. Using goto statement we can jump from top to bottom or bottom to top. To jump from one line to another line, the goto statement requires a label. Label is a name given to the instruction or line in the program. When we use a goto statement in the program, the execution control directly jumps to the line with the specified label.

### **Examples:**

#### **Program 1:**

```
#include
<stdio.h>void main()
{
int num,i=1;
printf("Enter the number whose table you want to print:");scanf("%d",&num);
table:printf("%dx%d=%d\n",num,i,num*i);i++;
if(i<=10)gototable
;
}
```

#### **Output:**

```
Enter the number whose table you want to print:10
10x1=10
10x2=20
10x3=30
10x4=40
10x5=50
10x6=60
10x7=70
10x8=80

10x9=90
10x10=100
```

#### **Program 2:**

```
#include
<stdio.h>void main()
{
int i,j,k;for(i=0;i<10;i++)
)
{
for(j=0;j<5;j++)
{
for(k=0;k<3;k++)
{
printf("%d %d
%d\n",i,j,k);if(j==3)
{
goto out;
}
```

```

}
}
}
out:
printf("cameoutoftheloop");
}

```

### Output:

```

000
001
002
010
011
012
020
021
022
030
cameoutoftheloop

```

### Programming Assignments using the above features.

#### Somemore Examples:

#### Program to Check Even or Odd

```

#include
<stdio.h>intmain()
{
intnum;
printf("Enter an integer:
");scanf("%d", &num);
if(num%2==0)//Trueifnumisperfectlydivisibleby2printf("%d
iseven.",num);
else
printf("%d is odd.", num);return0;
}

```

#### Output

Enter an integer: 77isodd.

#### Program to Check Vowel or consonant

```

#include
<stdio.h>intmain()
{
charc;
int lowercase_vowel,
uppercase_vowel;printf("Enteranalphabet:");
scanf("%c",&c);
//evaluatesto1ifvariablecisa lowercase vowel lowercase_vowel=(c=='a'||c==
'e'||c=='i'||c=='o'||c=='u');
// evaluatesto1ifvariablecisa uppercase vowel
uppercase_vowel=(c=='A'||c=='E'||c=='I'||c=='O'||c=='U');
//evaluates to 1(true)ifc is a vowel
if (lowercase_vowel ||

```

```
uppercase_vowel)printf("%c is a vowel.",c);
else
printf("%c is a consonant.", c);return0;
}
```

### **Output**

Enter an alphabet:  
G is a consonant.

### **Program to Check Leap Year**

```
#include
<stdio.h>intmain()
{
intyear;
printf("Enter a year:");

scanf("%d",&year);
// leap year if perfectly divisible by
400if(year%400==0)
{
printf("%d is a leap year.", year);
}
// not a leap year if divisible by 100
// but not divisible
by 400elseif(year% 100==0)
{
printf("%d is not a leap year.",year);
}
// leap year if not divisible by 100
// but divisible by 4else
if(year%4==0)
{
printf("%d is a leap year.", year);
}
// all other years are not leap year else
{
printf("%d is not a leap year.",year);
}
return0;
}
```

### **Output**

Enter a year: 1900  
1900 is not a leap year

### **Program to Check Alphabet**

```
#include
<stdio.h>intmain()
{
charc;
printf("Enter a character:
");scanf("%c",&c);
if((c>='a'&&c
<='z')||(c>='A'&&c<='Z'))printf("%c is an alphabet.",c);
else
printf("%c is not an alphabet.", c);return0;
}
```

### **Output**

Enter a character:\*

\* is not an alphabet

### To find the Factorial of a Number

```
#include
<stdio.h>int main()
{
int n,i;
unsigned long long fact=1;
printf("Enter an integer: ");

scanf("%d", &n);
// shows error if the user enters a negative integer if (n<0)
printf("Error! Factorial of a negative number doesn't exist.");else
{
for(i=1;i<=n;++i)
{
fact*=i;
}
printf("Factorial of %d=%llu",n,fact);
}
return 0;
}
```

#### Output

Enter an integer:

5  
Factorial of 10=120

### Multiplication Table Upto 10

```
#include
<stdio.h>int main()
{
int n,i;
printf("Enter an integer:");scanf("%d", &n);
for(i=1; i<=10;++i)
{
printf("%d* %d=%d\n", n,i, n*i);
}
return 0;
}
```

#### Output



Enteraninteger:9

9\*1= 9  
9\*2= 18  
9\*3= 27  
9\*4= 36  
9\*5= 45  
9\*6= 54  
9\*7= 63  
9\*8= 72  
9\*9= 81  
9\*10=90

### ToReverseanIntegernumber

```
#include
<stdio.h>intmain(){
int n,rev =0,remainder;printf("Enter
an integer: ");scanf("%d", &n);
while(n!= 0){

remainder= n% 10;
rev=rev*10+remainder;n/=10;
}
printf("Reversed number = %d",
rev);return0;
}
```

### Output

Enter an integer:  
2345Reversednumber=5432

### ProgramtoCheckPalindrome

```
#include
<stdio.h>intmain()
{
int n, reversedN = 0, remainder,
originalN;printf("Enteraninteger:");
scanf("%d", &n);originalN
=n;
//reversedinteger isstoredinreversedNwhile(n!=0){
remainder= n% 10;
reversedN= reversedN *10 + remainder;n/=10;
}
// palindrome if originalN and reversedN are
equalif(originalN==reversedN)
printf("%d is a palindrome.", originalN);else
printf("%d is not a palindrome.",
originalN);return0;
}
```

### Output

**Enter an integer:  
10011001isapalindrome.**

### Simple Calculator using switch Statement

```
#include
<stdio.h>intmain()
{
charoperator;doublefirst,second
;
printf("Enteran operator(+,-
*,):");scanf("%c",&operator);printf("Enter
twooperands:
");scanf("%lf%lf",&first,&second);switch(opera
tor)
{
case'+':printf("%.1lf+ %.1lf =%.1lf",first,second,first+ second);break;

case '-':printf("%.1lf - %.1lf = %.1lf", first, second, first - second);break;
case '*':printf("%.1lf * %.1lf = %.1lf", first, second, first * second);break;
case '/':printf("%.1lf / %.1lf = %.1lf", first, second, first / second);break;
default:printf("Error!operatorisnotcorrect");

}
return0;
}
```

### Output

**Enter an operator (+, -, \*):  
\*Entertwooperands:1.54.5  
1.5 \*4.5=6.75**

### Solved Questions

#### Short Answer Type Questions.

**Name of the first developer of C programming languages. Ans:-**

Dennis Ritchie

**What are the various types of statement available in C program? Ans:-**

Chasthreetypesofstatement.

- (i) Assignment=
- (ii) Selection (branching)if (expression)elseswitch.
- (iii) Iteration(looping)while(expression)for(expression;expression;expression)do{ block }

**Whatisthepurposeofi/ostatementin„C“? Ans:-**

- Itisusedtodisplayastringinputtedbygets()function.Itisalsousedtodisplayantext(message)onthescreen forprogramsimplicity.
- Input Output Statement.

**Explain the importance of C-language? Ans:-**

C is highly portable and is used for scripting system applications which form a major part of Windows, UNIX, and Linux operating system. C is a general-purpose programming language and can efficiently work on enterprise applications, games, graphics, and applications requiring calculations, etc.

### Long Answer Type Questions.

**Give the general structure of a C program, and discuss about each of the lines.**

*(2017-Winter)*

**Ans:-**

Programming in C is a difficult task for someone who is completely oblivious to the basic structure of a C program. After completing this tutorial, you would learn how the structure of C

Program looks like and soon you would become comfortable writing your own programs with ease!

Part of C program

- 1. # include <stdio.h>** – This command is a preprocessor directive in C that includes all standard input-output files before compiling any C program so as to make use of all those functions in our C program.
- 2. int main()** – This is the line from where the execution of the program starts. The main() function starts the execution of any program.
- 3. { (Opening bracket)** – This indicates the beginning of any function in the program (Here it indicates the beginning of the main function).
- 4. /\* some comments \*/** – Whatever is inside /\*———\*/ are not compiled and executed; they are only written for user understanding or for making the program interactive by inserting a comment line. These are known as multiline comments. Single line comments are represented with the help of 2 forward slashes —//———||.
- 5. printf(“HelloWorld”)** – The printf() command is included in the C stdio.h library, which helps to display the message on the output screen.
- 6. getch()** – This command helps to hold the screen.
- 7. return 0** – This command terminates the C program and returns a null value, that is, 0.
- 8. } (Closing brackets)** – This indicates the end of the function. (Here it indicates the end of the main function)

**WAP in C to find the real roots of a quadratic equation. (2017-Summer)**

**Ans:-**

```
#include
<stdio.h>#include
<math.h>int main()
{
inta,b,c,d;doubleroot1,root2;
printf("Enter a,b and c where a*x*x+b*x
+c=0\n");scanf("%d%d%d",&a,&b,&c);
d=b*b-4*a*c;if(d<0)
{
//complex roots, i is for iota(√-1, square root of -1)
printf("First root= %.2lf+ i%.2lf\n",-b/(double)(2*a), sqrt(-
d)/(2*a));printf("Second root= %.2lf-i%.2lf\n", -b/(double)(2*a),sqrt(-d)/(2*a));
}
else
{
//real roots
```

```

root1=(-b+sqrt(d))/(2*a);
root2=(-b-sqrt(d))/(2*a);printf("Firstroot
=%.2lf\n", root1);printf("Second root
=%.2lf\n",root2);
}
return0;
}

```

## **EXERCISE**

### **Short Answer Type Questions.**

Differentiate between Numeric and Character Constant?

How can you use a symbolic statement?

What do you mean by operator and operand?

What are the various types of Operator used in the C programming?

**(2016-Summer)**

Differentiate between logical and bitwise operator?

Differentiate between increment and decrement operator?

Differentiate between pre and post increment/decrement operator? **(2013-Summer)**

Differentiate between unary plus and unary minus operator?

Why conditional operator is called ternary operator?

Define Operator?

What is the relation of arithmetic operator with relational operator? Explain with an appropriate expression?

How can an expression with increment and decrement operators will be solved?

Differentiate between if-else and else-if statement?

Differentiate between ladder if and switch statement? **(2017-Summer)**

What do you mean by conditional control statement?

Define an iterative statement in a C program.

Give the general syntax of switch..... case statement in C.

Differentiate between do-while and while..... do statement in C.

### **Long Answer Type Questions**

WAP in C to print all 2-Digit Odd Numbers.

WAP in C to Calculate and print the factorial of a given number.

WAP in C to Compute and print the sum of the following series. **(2017-Winter)**

$S = 1 + 1/x + 1/x^2 + 1/x^3 + 1/x^4 \dots\dots\dots 1/x^n$

WAP in C to calculate the sum of the digits of a given number.

WAP in C to Compute and print the simple interest and compound interest.

WAP in C to compute  $(a+b)^2$ .

WAP in C to interchange value of two variables without using third variable.

WAP in C to find the sum of the given series. **(2015-Winter)**

$$1^n + 2^n + 3^n + 4^n + \dots + m^n$$

WAP in C to print.

\*  
\*\*  
\*\*\*

WAP in C to find the prime number. **(2017-Summer)**

WAP in C to find the greatest number among three integers. **(2015-Winter)**

WAP in C to compute and print the sum of the following series. **(2017-Winter)**

$$\square = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots + \frac{1}{\square 0}$$

WAP in C to find whether a number is Armstrong number or not.

\*\*\*\*\*

## CHAPTER-7:ADVANCEDFEATURESOFC

### Functions and Passing Parameters to the Function (Call by Value and Call by Reference)

#### Functions

- A function is a group of statements that together perform a task. Every C program has at least one function, which is main (), and all the most trivial programs can define additional functions.
- You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.
- A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.
- A function can also be referred to as a method or a sub-routine or a procedure, etc.

#### Defining a Function

- The general form of a function definition in C programming language is as follows—
  - return type function name (parameter list) {
  - body of the function
  - }
- A function definition in C programming consists of a function header and a function body. Here are all the parts of a function—
  - **Return Type** – A function may return a value. The return type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return type is the keyword void.
  - **Function Name** – This is the actual name of the function. The function name and the parameter list together constitute the function signature.
  - **Parameters** – A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as an actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.
  - **Function Body** – The function body contains a collection of statements that define what the function does.

#### Parameters in C functions

- A parameter is the symbolic name for "data" that goes into a function. There are two ways to pass parameters in C: Pass by Value, Pass by Reference.

#### Call by Value

- Pass by Value, means that a copy of the data is made and stored by way of the name of the parameter. Any changes to the parameter have NO effect on data in

the calling function.

- In call by value method, the value of the actual parameter is copied into the formal parameters. In other words, we can say that the value of the variable is used in the function call in the call by value method.

- In call by value method, we cannot modify the value of the actual parameter by the formal parameter.

- In call by value, different memory is allocated for actual and formal parameters since the value of the actual parameter is copied into the formal parameter.

- The actual parameter is the argument which is used in the function call whereas formal parameter is the argument which is used in the function definition.

### **Call by Reference**

- A reference parameter "refers" to the original data in the calling function. Thus, any changes made to the parameter are also made to the original variable.

- In call by reference, the address of the variable is passed into the function call as the actual parameter.

- The value of the actual parameters can be modified by changing the formal parameters since the address of the actual parameters is passed.

- In call by reference, the memory allocation is similar for both formal parameters and actual parameters. All the operations in the function are performed on the value stored at the address of the actual parameters, and the modified value gets stored at the same address.

- There are two ways to make a pass by reference parameter:

### **ARRAYS**

- Arrays are always pass by reference in C. Any change made to the parameter containing the array will change the value of the original array.

**The ampersand (&) used in the function prototype.** Function(&parameter name)

- To make a normal parameter into a pass by reference parameter, we use the "&param" notation. The ampersand (&) is the syntax to tell C that any changes made to the parameter also modify the original variable containing the data.

### **Call by Value Example: Swapping the values of the two variables (Swapping not Possible)**

```
#include<stdio.h>
void swap(int , int); //prototype of the function
int main()
{
int a = 10; int b = 20;
printf("Before swapping the values in main a = %d, b = %d\n", a, b); swap(a, b);
printf("After swapping values in main a = %d, b = %d\n", a, b);
//The value of actual parameters do not change by changing the formal parameters in call by value, a=10, b=20
}

void swap(int a, int b)
{
int temp; temp = a;
```

```
a=b;b=temp;
printf("Afterswappingvaluesinfunctiona=%d,b=%d\n",a,b);//Formalparameters,a
=20,b=10
}
```

### Output

**Before swapping the values in main a = 10, b = 20**  
**After swapping values in function a = 20, b = 10**  
**After swapping values in main a=10,b =20**

### Call by reference Example: Swapping the values of the two variables

```
#include<stdio.h>
void swap(int *, int *); //prototype of the function
int main()
{
int a = 10;int b=20;
printf("Before swapping the values in main a=%d,b=%d\n",a,b);//printing the value of a and b in main
swap(&a,&b);
printf("Afterswapping values in main a=%d,b=%d\n",a,b);//The values of actual parameters do change in call by
reference,a=10,b=20
}
void swap (int*a, int*b)
{
int temp;temp=*a;
*a=*b;
*b=temp;
printf("Afterswapping values in function a=%d,b=%d\n",*a,*b);//Formal parameters,a=20,b=10
}
```

### Output:

**Before swapping the values in main a = 10, b = 20**  
**After swapping values in function a = 20, b = 10**  
**After swapping values in main a =20,b =10**

## Scope of Variables and Storage Classes, Recursion, Function and Types of Recursion

### Scope of variables

- When we declare a variable in a program, it cannot be accessed against the scope rules. Variables can be accessed based on their scope. The scope of a variable decides the portion of a program in which the variable can be accessed. The scope of the variable is defined as follows...
- Scope of a variable is the portion of the program where a defined variable can be accessed.
- The variable scope defines the visibility of variable in the program. Scope of a variable depends on the position of variable declaration.
- In C programming language, a variable can be declared in three different positions and they are as follows..
  - Before the function definition (Global Declaration)
  - Inside the function or block (Local Declaration)
  - In the function definition parameters (Formal Parameters)

### Before the function definition (Global Declaration)

### Example Program



```

#include<stdio.h>#include
<conio.h>intnum1,num2;v
oidmain()
{
voidaddition()
;voidsubtraction();
void multiplication()
;clrscr();
num1 = 10
;num2=20;
printf("num1= %d,num2=%d",num1,num2) ;addition() ;
subtraction()
;multiplication()
;getch() ;
}
void addition()
{
intresult;
result=num1+num2;printf("\naddition
=%d",result);
}

voidsubtraction()
{
intresult;
result=num1-num2;printf("\nsubtraction=%d",
result);
}
voidmultiplication()
{
intresult;
result=num1*num2;printf("\nmultiplication=%
d",result);
}

```

### **Output:**

Inside thefunctionor block(Local Declaration)

### **ExampleProgram**

```

#include<stdio.h>

#include<conio.h>voidmai
n()
{
voidaddition() ;int num1,
num2 ;clrscr();
num1 = 10
;num2=20;
printf("num1= %d,num2=%d",num1,num2) ;addition() ;
getch() ;
}
void addition()
{

```

```

intsumResult ;
sumResult=num1+num2
;printf("\naddition=%d",sumResult);
}

```

### Output:

Inthefunctiondefinitionparameters (FormalParameters)

### ExampleProgram

```

#include<stdio.h>#include
<conio.h>voidmain()
{
void addition(int, int)
;intnum1,num2;clrscr();
num1 = 10
;num2=20;
addition(num1, num2) ;getch() ;
}
void addition(int a,intb)
{
intsumResult;sumResult= a+b;
printf("\naddition =%d",sumResult);
}

```

### StorageClasses

- StorageclassesinCareusedtodeterminethelifetime,visibility,memorylocation,andinitialvalueofavariable. Thereare four typesofstorageclassesin C

- Automatic
- External
- Static
- Register

#### Automatic

- Automaticvariables areallocatedmemoryautomaticallyat runtime.
- Thevisibilityof theautomaticvariables islimited to theblock in which theyaredefined.
- Thescopeoftheautomaticvariables islimited tothe blockin whichtheyaredefined.
- Theautomaticvariables areinitialized togarbage bydefault.
  - Thememoryassignedtoautomatic variablesgetsfreedupon exitingfrom theblock.
- The keyword usedfordefiningautomatic variablesisauto.
- Everylocal variableis automatic in C bydefault.

### Example

```

#include
<stdio.h>intmain()
{
int a;
//autocharb;floatc;
printf("%d %c %f",a,b,c); // printing initial default value of automatic variables a, b, and c.return0;
}

```

### Output:

garbagegarbagegarbage

## Static

- The variables defined as static specifier can hold their value between the multiple function calls.
- Static local variables are visible only to the function or the block in which they are defined.
- A same static variable can be declared many times but can be assigned at only one time.
- Default initial value of the static integral variable is 0 otherwise null.
- The visibility of the static global variable is limited to the file in which it has declared.
- The keyword used to define static variable is static.

## Example

```
#include <stdio.h>
static int i;
static float f;
static char s[100];
void main()
{
    printf("%d %d %f %s", c, i, f); // the initial default value of c, i, and f will be printed.
}
```

### Output:

0 0 0.000000 (null)

## Register

- The variables defined as the register is allocated the memory into the CPU registers depending upon the size of the memory remaining in the CPU.
- We cannot dereference the register variables, i.e., we cannot use & operator for the register variable.
- The access time of the register variables is faster than the automatic variables.
- The initial default value of the register local variables is 0.
- The register keyword is used for the variable which should be stored in the CPU register. However, it is compiler's choice whether or not; the variables can be stored in the register.
- We can store pointers into the register, i.e., a register can store the address of a variable.
- Static variables cannot be stored into the registers since we cannot use more than one storage specifier for the same variable.

## Example

```
#include <stdio.h>
int main()
{
    register int a; // variable is allocated memory in the CPU register. The initial default value of a is 0.
    printf("%d", a);
}
```

### Output:

0

## External

- The external storage class is used to tell the compiler that the variable defined as extern is declared with an extern linkage elsewhere in the program.
- The variables declared as extern are not allocated any memory. It is only declaration and intended to specify that the variable is declared elsewhere in the program.
- The default initial value of external integral type is 0 otherwise null.
- We can only initialize the extern variable globally, i.e., we cannot initialize the external variable within an

y block or method.

- An external variable can be declared many times but can be initialized only once.
- If a variable is declared as external then the compiler searches for that variable to be initialized somewhere in the program which may be extern or static. If it is not, then the compiler will show an error.

### Example

```
#include
<stdio.h>
int main()
{
extern int
a;
printf("%d", a);
}
```

### Output

```
main.c:(.text+0x6): undefined reference to `a'
collect2:
error: ld returned 1 exit status
```

### Recursion Function

- Recursion is the process of repeating items in a self-similar way. In programming languages, if a program allows you to call a function inside the same function, then it is called a recursive call of the function.

```
void recursion()
{
recursion (); /*function calls itself*/
}
int main()
{recursion();
}
```

- The C programming language supports recursion, i.e., a function to call itself. But while using recursion, programmers need to be careful to define an exit condition from the function, otherwise it will go into an infinite loop.
- Recursive functions are very useful to solve many mathematical problems, such as calculating the factorial of a number, generating Fibonacci series, etc.

### Types of Recursion

Recursion remains of two types depending on whether a function calls itself from within itself or whether two functions call each other mutually.

Thus, the two types of recursion are:

- Direct recursion
- Indirect recursion

Recursion may be further categorized as:

- Linear recursion
- Binary recursion
- Multiple recursion

- Conceptually you can think of a one-dimensional array as a row, where elements are stored one after another.
- Syntax: `data type array_name[size];`
- datatype: It denotes the type of the elements in the array.
- array\_name: Name of the array. It must be a valid identifier.
- size: Number of elements an array can hold. Here are some examples of array declarations:

`int num[100]; float`

`temp[20]; char ch[50];`

### Multidimensional Arrays

- The simplest form of multidimensional array is the two-dimensional array. A two-dimensional array is, in essence, a list of one-dimensional arrays. To declare a two-dimensional integer array of size `[x][y]`, you would write something as follows—  
`type array Name[x ][y ];`
- Where type can be any valid C data type and array Name will be a valid C identifier.
- A two-dimensional array can be considered as a table which will have x number of rows and y number of columns.
- A two-dimensional array, which contains three rows and four columns, can be shown as follows—

	Column 0	Column 1	Column 2	Column 3
Row 0	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
Row 1	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
Row 2	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>

- Thus, every element in the array `a` is identified by an element name of the form `a[i ][ j ]`, where 'a' is the name of the array, and 'i' and 'j' are the subscripts that uniquely identify each element in 'a'.

### String operations

- Strings are actually one-dimensional array of characters terminated by a null character `'\0'`. Thus a null-terminated string contains the characters that comprise the string followed by a null.
- The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

```
char greeting[6]={'H','e','l','l','o','\0'};
```

- If you follow the rule of array initialization then you can write the above statement as follows—  
`char greeting[] = "Hello";`
- Following is the memory presentation of the above defined string in C/C++—
- Actually, you do not place the null character at the end of a string constant. The C compiler automatically places the `'\0'` at the end of the string when it initializes the array.

Let us try to print the above mentioned string – Live Demo

```
#include <stdio.h>
int main()
{
```

### Pointers

```
    char greeting[6]={'H','e','l','l','o','\0'}; printf("Greeting message: %s\n", greeting ); return 0;
```

□ A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. The general form of a pointer variable declaration is—

type\*var-name;

□ Here, type is the pointer's base type; it must be a valid C data type and var-name is the name of the pointer variable. The asterisk \* used to declare a pointer is the same asterisk used for multiplication.

### How to Use Pointers?

□ There are a few important operations, which we will do with the help of pointers very frequently.

□ We define a pointer variable,

□ assign the address of a variable to a pointer and

□ Finally access the value at the address available in the pointer variable.

□ This is done by using unary operator \* that returns the value of the variable located at the address specified by its operand. The following example makes use of these operations

```
—
#include <stdio.h>
int main() {
int var = 20; /* actual variable declaration */
int *ip; /* pointer variable declaration */
ip = &var; /* store address of var in pointer variable */
printf("Address of var variable: %x\n", &var);
/* address stored in pointer variable */
printf("Address stored in ip variable: %x\n", ip);
/* access the value using the pointer */
printf("Value of *ip variable: %d\n", *ip);
return 0;
}
```

### **Pointer Expression and Pointer Arithmetic Programming, Assignments using the above features.**

• Pointers are used to point to the address of a variable. A pointer is declared by preceding the name of the pointer by an asterisk (\*).

• Syntax:

data type \* pointer\_name;

• When we need to initialize a pointer with variable's location, we use ampersand sign (&) before the variable name.

// Declaration of integer variable int var = 10;

// Initialization of pointer

variable int \* pointer = &var;

➤ The ampersand (&) is used to get the address of a variable. We can directly find the location of any identifier by just preceding it with an ampersand (&) sign.

### **Example:**

// This code prints the address of x #include <stdio.h>

```
int main()
{
int x = 10;
// Prints address of x
printf("Address of variable x = %p", &x);
return 0;
}
```

## Pointer Arithmetic Programming

Pointer is an address, which is a numeric value. Therefore, you can perform arithmetic operations on a pointer just as you can on a numeric value. There are four arithmetic operators that can be used on pointers: ++, --, +, and -

To understand pointer arithmetic, let us consider that ptr is an integer pointer which points to the address 1000. Assuming 32-bit integers, let us perform the following arithmetic operation on the pointer – ptr++

After the above operation, the ptr will point to the location 1004 because each time ptr is incremented, it will point to the next integer location which is 4 bytes next to the current location. This operation will move the pointer to the next memory location without impacting the actual value at the memory location. If ptr points to a character whose address is 1000, then the above operation will point to the location 1001 because the next character will be available at 1001.

## Incrementing a Pointer

We prefer using a pointer in our program instead of an array because the variable pointer can be incremented, unlike the array name which cannot be incremented because it is a constant pointer. The following program increments the variable pointer to access each succeeding element of the array –

```
#include <stdio.h>
const int MAX = 3;
void main()
{
    int var[] = {10, 100, 200};
    int i, *ptr;
    ptr = var;
    /*let us have array address in pointer*/
    for(i=0; i<MAX; i++)
    {
        printf("Address of var[%d] = %x\n", i, ptr);
        printf("Value of var[%d] = %d\n", i, *ptr);
        ptr++;
        /*move to the next location*/
    }
}
```

**Output**

```
Addressofvar[0]=  
bf882b30Valueofvar[0]=10  
Addressofvar[1]=  
bf882b34Valueofvar[1]=  
100Addressofvar[2] =  
bf882b38Valueofvar[2]=200
```

## Decrementing a Pointer

The same considerations apply to decrementing a pointer, which decreases its value by the number of bytes of its data type as shown below –

```
#include <stdio.h>const int MAX  
= 3;intmain()  
{  
  
intvar[] = { 10, 100, 200 };  
inti, *ptr;  
ptr = &var[MAX-1];/* let us have array address in pointer */for(i=MAX; i>0; i--)  
{  
printf("Addressofvar[%d]= %x\n", i-1, ptr);printf("Value  
ofvar[%d]= %d\n", i-1, *ptr );ptr--;/* move to the previous  
location */  
}  
return 0;  
}
```

### Output

```
Addressofvar[2]=  
bfedbcd8Valueofvar[2]=200Address  
ofvar[1]=  
bfedbcd4Valueofvar[1]=100Address  
ofvar[0]=  
bfedbcd0Valueofvar[0]=10
```

## Pointer Comparisons

Pointers may be compared by using relational operators, such as ==, <, and >. If p1 and p2 point to variables that are related to each other, such as elements of the same array, then p1 and p2 can be meaningfully compared.

The following program modifies the previous example – one by incrementing the variable pointer so long as the address to which it points is either less than or equal to the address of the last element of the array, which is &var[MAX-1] –

### Example

```
#include <stdio.h>const int MAX  
= 3;intmain()  
{  
intvar[] = { 10, 100, 200 };  
inti, *ptr;  
/* let us have address of the first element in pointer */ptr=var;  
i=0;  
while( ptr<=&var[MAX-1]){  
printf("Addressofvar[%d]=% x\n", i, ptr);
```



```
printf("Value ofvar[%d]= %d\n", i,*ptr );
/* pointtothenextlocation*/ptr++;
i++;
}
return 0;
}
```

## Output

```
Addressofvar[0]=
bfdcb20Valueofvar[0]=10
Addressofvar[1]= bfdcb24

Valueofvar[1]=100Addressofvar[2]=
bfdcb28Valueofvar[2]=200
```

## Structure and Union (Only concepts, No Programming)

### What is a structure?

A structure is a user-defined data type in C. A structure creates a data type that can be used to group items of possibly different types into a single type.

### How to create a structure?

—**struct** keyword is used to create a structure.

### Following is an example

```
struct address
{
char name[50];

char street[100]; char city[50]; char
state[20]; int pin;
};
```

### How to declare structure variables?

- A structure variable can either be declared with structure declaration or as a separate declaration like basic types.
- A variable declaration with structure

```
declaration.structPoint
```

```
{
int x,y;
}p1;
```

The variable p1 is declared with 'Point'.

- A variable declaration like basic data types

```
struct Point
{
int x,y;
};
int main()
{
struct Point p1; //The variable p1 is declared like a normal variable
}
```

## Union

- A union is a special data type available in C that allows storing different data types in the same memory location.
- You can define a union with many members, but only one member can contain a value at any given time.
- Unions provide an efficient way of using the same memory location for multiple purposes.

### Defining a Union:

- To define a union, you must use the union statement in the same way as you did while defining a structure.
- The union statement defines a new data type with more than one member for your program.
- The format of the union statement is as follows:

```
union[unionname]
{
member
definition;memberdefinition;
...
memberdefinition;
};
```

	STRUCTURE	UNION
<b>Keyword</b>	The keyword <b>struct</b> is used to define a structure	The keyword <b>union</b> is used to define a union.
<b>Size</b>	When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is <b>greater than or equal to the sum of sizes of its members.</b>	when a variable is associated with a union, the compiler allocates the memory by considering the size of the largest memory. So, size of union is <b>equal to the size of largest member.</b>
<b>Memory</b>	Each member within a structure is assigned unique storage area of location.	Memory allocated is shared by individual members of union.
<b>Value Altering</b>	Altering the value of a member will not affect other members of the structure.	Altering the value of any of the member will alter other member values.
<b>Accessing members</b>	Individual member can be accessed at a time.	Only one member can be accessed at a time.
<b>Initialization of Members</b>	Several members of a structure can initialize at once.	Only the first member of a union can be initialized.

### Standard Library Functions

Many basic housekeeping functions are available to a C program in form of standard library functions. To call these, a program must `#include` the appropriate `.h` file. Most compilers link in the standard library code by default. The functions listed in the next section are

the most commonly used ones, but there are many more which are not listed here.

	file input and output
<code>ctype.h</code>	character tests
<code>string.h</code>	string operations
<code>math.h</code>	mathematical functions such as <code>sin()</code> and <code>cos()</code>
<code>stdlib.h</code>	utility functions such as <code>malloc()</code> and <code>rand()</code>
	<code>assert()</code> debugging macro
<code>stdarg.h</code>	support for functions with variable numbers of arguments
<code>setjmp.h</code>	support for non-local flow control jumps
<code>signal.h</code>	support for exceptional conditions
<code>time.h</code>	date and time

limits.h, float.h constants which define type range values such as INT\_MAX

## Examples of advance programming in C

### Example of Array in C programming to find out the average of 4 integers

```
#include <stdio.h>
int main()
{
    int avg = 0;
    int sum = 0;
    int x = 0;
    int num[4]; /* Array-declaration-length 4 */
    for (x=0; x<4; x++) /* We are using for loop to traverse through the */
    {
        printf("Enter number %d \n", (x+1));
        scanf("%d", &num[x]);
    }
    for (x=0; x<4; x++)
    {
        sum = sum + num[x];
    }
    avg = sum / 4;
    printf("Average of entered number is: %d", avg);
    return 0;
}
```

#### Output:

```
Enter number 1
10
Enter number 2
10
Enter number 3
20
Enter number 4
40
Average of entered number is: 20
```

### Example to print the address of array elements

```
#include <stdio.h>
int main()
{
    int val[7] = { 11, 22, 33, 44, 55, 66, 77 };
    /* for loop to print value and address of each element of array */
    for (int i = 0; i < 7; i++)
    {
        printf("val[%d]: value is %d and address is %d\n", i, val[i], &val[i]);
    }
    return 0;
}
```

#### Output:

```
val[0]: value is 11 and address is 1423453232
val[1]: value is 22 and address is 1423453236
val[2]: value is 33 and address is 1423453240
val[3]: value is 44 and address is 1423453244
val[4]: value is 55 and address is
```

1423453248val[5]: value is 66 and address is  
1423453252val[6]: value is 77 and address is 1423453256

### Example: Passing Pointer to a Function in C

```
#include<stdio.h>
void salaryhike(int*var, intb)
{
*var=*var+b;
}
intmain()
{
```

#### Output:

```
intsalary=0,bonus=0;
printf("Enter the employee current
salary:");scanf("%d",&salary);
printf("Enterbonus:");scanf("%d",&bonus);salaryhi
ke(&salary,bonus);printf("Final salary: %d",
salary);return 0;
}
Enter the employee current
salary:10000Enterbonus:2000
Finalsalary: 12000
```

### Example for Swapping two numbers using Pointers

```
#include<stdio.h>
voidswapnum(int*num1,int*num2)
{
inttempnum;
tempnum= *num1;
*num1= *num2;
*num2= tempnum;
}
intmain()
{
intv1=11,v2=77;printf("Before
swapping:");printf("\nValue of v1 is: %d",
v1);printf("\nValue of v2 is: %d", v2);
/*calling swap function*/swapnum(&v1,&v2
);printf("\nAfter swapping:");printf("\nValue of v1
is: %d", v1);printf("\nValue of v2 is: %d",v2);
}
```

#### Output:

```
Before swapping:
Value of v1 is: 11 Value
of v2 is: 77 After
swapping:
```

Value of v1 is:  
77 Value of v2 is: 11

### Example: Program to find the size of an array

```
#include <stdio.h>
int main()
{
    double arr[] = {11, 22, 33, 44, 55, 66};
    int n;
    /* Calculating the size of the array with this formula.
    *          n = sizeof(array_name) / sizeof(array_name[0])
    *          This is a universal formula to find number of elements in
    *          an array, which means it will work for arrays of all data
    *          types such as int, char, float etc.
    */
    n = sizeof(arr) / sizeof(arr[0]);
    printf("Size of the array is: %d\n", n);
    return 0;
}
```

#### Output:

Size of the array is: 6

### Example: Program to Calculate Standard Deviation

```
#include
<math.h> #include <stdio.h>
float calculateSD(float data[]);
int main()
{
    int i;
    float data[10];
    printf("Enter 10 elements:");

    for(i=0; i<10; ++i) scanf("%f", &data[i]);
    printf("\n Standard Deviation = %.6f", calculateSD(data));
    return 0;
}

float calculateSD(float data[])
{
    float sum = 0.0, mean, SD = 0.0;
    int i;
    for(i=0; i<10; ++i)
    {
        sum += data[i];
    }
    mean = sum / 10;
    for(i=0; i<10; ++i)
    SD += pow(data[i] - mean, 2);
    return sqrt(SD / 10);
}
```

#### Output

Enter 10 elements: 1  
2  
3  
4  
5

6  
7  
8  
9  
10  
StandardDeviation =2.872281

### Example:ProgramtoAddTwoMatrices

```
#include <stdio.h>intmain()
{
int r,c, a[100][100],b[100][100], sum[100][100], i, j;
printf("Enter the number of rows (between 1 and 100): ");scanf("%d", &r);
printf("Enter the numberofcolumns (between1and100):");
scanf("%d",&c);
printf("\nEnter elements of 1st matrix:\n");for(i=0;i<r;++i)
for(j=0;j<c;++j)
{
printf("Enterelementa%d%d:", i +1,j+ 1);scanf("%d",&a[i][j]);
}

printf("Enterelementsof2ndmatrix:\n");
for(j=0;j<c; ++j)
{
printf("Enter element a%d%d: ", i + 1, j + 1);scanf("%d", &b[i][j]);
}
for (i = 0;i<r; ++i)//adding twomatricesfor(j=0;j<c;++j)
sum[i][j]= a[i][j] + b[i][j];
printf("\nSumoftwomatrices: \n"); // printing the
resultfor(i=0;i<r;++i)
for(j=0;j<c;++j)
{
printf("%d", sum[i][j]);if(j==c-1)
{
printf("\n\n");
}
}
return0;
}
```

### Output

Enterthenerofrows(between1and100):2Enter the number of  
columns (between 1 and 100): 3Enterelementsof1stmatrix:  
Enter element a11: 2Enter  
element a12: 3Enter element  
a13: 4Enter element a21:  
5Enter element a22:  
2Enterelementa23:3  
Enter elements of 2nd  
matrix:Enterelementa11:-4  
Enter element a12: 5Enter  
element a13: 3Enter element

a21: 5Enter element a22:  
6Enter element a23:  
3Sumoftwomatrices:

-2 8 7  
108 6

### Example: Multiply Matrices by Passing it to a Function

```
#include<stdio.h>
// function to get matrix elements entered by the user
void getMatrixElements(int matrix[][10], int row, int column)
{
    printf("\nEnter elements:\n");
        for (int i=0; i<row; ++i)
        {
            for (int j=0; j<column; ++j)
            {
                printf("Enter a%d%d: ", i+1, j+1); scanf("%d", &matrix[i][j]);
            }
        }
}

// function to multiply two matrices
void multiplyMatrices(int first[][10], int second[][10], int result[][10], int r1, int c1, int r2, int c2)
{
    // Initializing elements of matrix mult to 0.
    for (int i=0; i<r1; ++i)
    {
        for (int j=0; j<c2; ++j)
        {
            result[i][j]=0;
        }
    }

    // Multiplying first and second matrices and storing it in result
    for (int i=0; i<r1; ++i) {
        for (int j=0; j<c2; ++j) {
            for (int k=0; k<c1; ++k) {
                result[i][j] += first[i][k] * second[k][j];
            }
        }
    }

// function to display the matrix
void display(int result[][10], int row, int column)
{ printf("\nOutput Matrix:\n");
    for (int i=0; i<row; ++i) {
        for (int j=0; j<column; ++j) { printf("%d", result[i][j]);
            if (j == column - 1) printf("\n");
```

```

}
}
}

int main()
{
int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2; printf("Enter
rows and column for the first matrix:"); scanf("%d%d",&r1,&c1);
printf("Enter rows and column for the second matrix: "); scanf("%d%d",&r2,&c2);
//Taking input until
// 1st matrix column is not equal to 2nd matrix row while(c1!=r2)
{
printf("Error! Enter rows and columns again.\n"); printf("Enter rows and columns for
the first matrix: "); scanf("%d%d", &r1,&c1);
printf("Enter rows and columns for the second matrix: "); scanf("%d%d",
&r2,&c2);
}
    r1,c1,r2, c2); //multiply two matrices. display(result,r1,c2); //display the result
return 0;
}

```

### Output

```

getMat
rixEle
ments(
first,r1,
c1); //g
etelem
entsoft
he first
matrix
getMat
rixEle
ments(
second
,r2,c2);
//getele
mentso
fthese
condma
trixmul
tiplyM
atrices(
first,
second
,result,

```



Enter rows and column for the first matrix: 2 3  
Enter rows and column for the second matrix: 3 2

Enter  
elements:  
Enter a11: 2  
Enter a12: -3  
Enter a13: 4  
Enter a21: 5  
Enter a22: 3  
Enter a23: 5

Enter  
elements:  
Enter a11: 3  
Enter a12: 3  
Enter a21: 5  
Enter a22: 0

Enter a31: -3  
Enter a32: 4

Output Matrix:  
-2 1 2 2

15 9 1 7 9

### Example: Program to Find the Transpose of a Matrix

```
#include <stdio.h>
int main()
{
    int a[10][10], transpose[10][10], r, c, i, j;
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);
    // Assigning elements to the matrix
    printf("\nEnter matrix elements:\n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {
            printf("Enter element a%d%d:", i+1, j+1);
            scanf("%d", &a[i][j]);
        }

    // Displaying the matrix a[][]

    printf("\nEntered matrix: \n");
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {
            printf("%d", a[i][j]);
            if(j==c-1) printf("\n");
        }

    // Finding the transpose of matrix A
    for(i=0; i<r; ++i)
        for(j=0; j<c; ++j)
        {
            transpose[j][i] = a[i][j];
        }

    // Displaying the transpose of matrix a
    printf("\nTranspose
```

```

of the matrix:\n");for(i=0;i<c;++i)
for(j=0; j<r;++j)

{
printf("%d", transpose[i][j]);if(j==r-1)
printf("\n");
}
return0;
}

```

### Output

Enter rows and columns:23

Enter matrix elements:

Enter element a11:

1Enter element a12:

4Enter element a13: 0Enter

element a21: -

5Enter element a22:

2Enter element a23: 7

Entered matrix:

140

-527

Transpose of the matrix:

1-5

42

0            7

### Using Call by Reference

```
#include<stdio.h>
```

```
void cyclicSwap(int *a, int *b, int *c);int main()
```

```
{
```

```
int a,b,c;
```

```
printf("Enter a, b and c respectively: ");scanf("%d
```

```
%d%d",&a,&b,&c);printf("Value before swapping:\n");
```

```
printf("a= %d\nb=%d\nc = %d\n", a,b,c);cyclicSwap(&a,&b,&c);
```

```
printf("Value after swapping:\n");
```

```
printf("a=%d\nb=%d\nc=%d",a, b,c);return0;
```

```
}
```

```
void cyclicSwap(int *n1,int *n2, int*n3)
```

```
{
```

```
int temp;
```

```
//swapping in cyclic order
```

```
temp=*n2;
```

```
*n2 = *n1;
```

```
*n1 = *n3;
```

```
*n3=temp;
```

```
}
```

### Output

Enter a, b and c respectively:12

3

Value before swapping:a=1

b=2

c=3

Value after swapping:a=3

b=1

c=2

## Solved Questions

### Short Answer Type Questions.

**What is an array? What is its importance in C? (2015-Summer)**

**Ans:-**

Arrays are a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

**What is a single dimensional array? How can it be initialized? Ans:-**

1D Arrays in C language—A list of items can be given one variable name using only one subscript and such a variable is called a single subscripted variable or one dimensional array.

**What are the various string operations? Ans:-**

strcat-concatenate two strings. strchr-string scanning operation. strcmp-compare two strings. strcpy-copy a string.

**How can you reverse a string?**

**Ans:-** C program to reverse a

string in main()

```
{  
chars[100];  
printf("Enter a string to reverse\n"); gets(s); strrev(s);  
printf("Reverse of the string: %s\n", s); return 0;  
}
```

### Long Answer Type Questions.

**Write a program to find the sum of even numbers of an array? (2017-Summer)**

**Ans:**

```
#include <stdio.h> main()  
{  
int a[10], i, sum=0;  
printf("Enter upto 5 Values:  
"); for(i=0; i<5; i++) scanf("%d", &a[i]);  
for(i=0; i<5; i++)  
{  
if(a[i]%2==0) sum=sum+a[i];  
}  
printf("Total Sum of Even values is: %d ", sum); getch();  
}
```

**Output:**

Enter upto 5 Values: 2 3 5 4 7  
Total Sum of Even values is: 6

**Write a program which prints the smallest element?Ans:**

```
#include<stdio.h>

intmain()
{
int a[30],i, num,smallest;printf("\nEnter no of elements
:");scanf("%d",&num);
//Readnelementsinan arrayfor(i=0;i<num;i++)
scanf("%d",&a[i]);
//Consider first element as smallestsmallest=a[0];
for (i=0;i< num;i++)
{
if(a[i] <smallest)
{
smallest =a[i];
}
}
//Printoutthe Result
printf("\nSmallest Element : %d", smallest);return(0);
}
```

**Output:**

Enterno ofelements : 51144225599

Smallest Element:11

**ShortAnswerTypeQuestions.**

### **EXERCISE**

Whatisamulti-dimensionalarray?Howisitinitialized?(2017-Summer)

Howcanatwo-dimensional arraybedeclaredinC program?

Whatisbuilt-infunction? Give someexampleofthesefunctions?

Give someexamplesofFunctionPrototypes?

Differentiate betweenstrcat( ) andstrcpy( )?

Howcan youapplyarithmetic operations onstrings?

Howis astructuredifferentor similarfrom/toanarray?

Howcanyoucreatestructure variables?(2015-Summer)

Howcanyoudefinearraywithinstructres?

Defineaunion? (2014-Winter)

Howcanyoudeclare a union?

Definea pointer?(2015-Winter)(2016-Winter)

Whatis the relationbetween pointers and arrays?(2015-Summer)

Whatis thefunction ofmalloc() libraryfunction?

Whatis the meaningofreference?

Whatisaddressoperator?

Give an example of pointer initialization?

Briefly explain the association between one-dimensional arrays and pointers?

Explain the relation between pointers and functions?

### **Long Answer Type Questions**

Write a program which replaces every blank space in a string with underscore?

How can you calculate the size of an array? Illustrate with the help of example. (2014-Summer)

How we can call a function by

i. Reference

ii. Value

Write a C program which illustrates passing of arguments by reference? (2016-Summer)

Write a program which reads a string and prints the number of words in it?

What is the difference between `putc()` and `puts()` and `putchar()` functions? (2016-Winter)

\*\*\*\*\*

### **Chapter-wise Multiple Choice Questions**

#### **CHAPTER-1: MCO**

The "0" and "1" in the binary numbering system are called Binary digits or known as:

- a) Bytes
- b) Kilobytes
- c) Bits
- d) Kilobits

**Ans: c) Bits**

The generation based on VLSI microprocessor:

- a) 1st
- b) 2nd
- c) 3rd
- d) 4th

**Ans: d) 4th**

ULSI stands for?

- a) Ultra Large Scale Integration
- b) Under Lower Scale Integration
- c) Ultra Lower Scale Integration
- d) Under Large Scale Integration

**Ans: a) Ultra Large Scale Integration**

The period of \_\_\_\_\_ generation was 1952-1964.

- a) 1st
- b) 2nd

- c) 3rd
- d) 4th

**Ans:b) 2nd**

**The brain of any computer system is:**

- a) ALU
- b) Memory
- CPU
- c) Control Unit

**Ans:c) CPU**

**CD-ROM is a:**

- a) Semiconductor memory
- b) Memory register
- c) Magnetic memory
- d) None of the above

**Ans:d) None of the above**

**A hybrid computer:**

- a) Resembles digital computer
- b) Resembles analogue computer
- c) Resembles both a digital and an analogue computer
- d) None of the above

**Ans:c) Resembles both a digital and an analogue computer**

**What was the computer invented by Atanasoff and Clifford?**

- a) Mark I
- b) ABC
- c) Z3
- d) None of the above

**Ans:b) ABC**

**Which of the following is not an input device?**

- a) OCR
- b) Optical scanners
- c) Voice recognition device
- d) COM (Computer Output to Microfilm)

**Ans:d) COM (Computer Output to Microfilm)**

**When was the vacuum tube invented?**

- a) 1900
- b) 1906
- c) 1910
- d) 1880

**Ans:b) 1906**

**What does part number, part description and number of parts ordered belong to?**

- a) Output
- b) Input
- c) Feedback
- d) Control

**Ans: b) Input**

**CPU is mainly responsible for:**

- a) Calculations
- b) Processing the data
- c) Both 1 and 2
- d) Neither 1 nor 2

**Ans: c) Both 1 and 2**

**What is the use of control unit of a microprocessor?**

- a) To accept input data from keyboard
- b) To perform arithmetic and logic functions
- c) To store data in memory
- d) All of the above

**Ans: b) To perform arithmetic and logic functions**

**In the CPU of a computer, the logical unit is mainly responsible for :**

- a) Control flow of information
- b) Comparing numbers
- c) Producing result
- d) Mathematical operation

**Ans: b) Comparing numbers**

**CPU is made up of two main components, and they are :**

- a) Registers and main memory
- b) Control unit and registers
- c) ALU and bus
- d) Control unit and ALU

**Ans: d) Control unit and ALU**

**A bit is the measuring unit of the width of a processor's data path. Commonly used data path is of:**

- a) 24 bits
- b) 32 bits
- c) 8 bits
- d) 16 bits

**Ans: c) 8 bits**

**What is the combination with which central processing unit is made ?**

- a) Arithmetic logic and control unit
- b) Control and storage
- c) Control and output unit
- d) All of the above

**Ans: a) Arithmetic logic and control unit**

**Which language is the set of rules that tell the computer what operation to perform?**

- a) Programming language
- b) Command language
- c) Procedural language
- d) Structures

**Ans: a) Programming language**

**Which one of the following statements is true for the machine language?**

- a) Programs were first written in this language
- b) Computer understands only this language
- c) It differs from computer to computer
- d) All of the above

**Ans: d) All of the above**

**The language which is understood by the computer is:**

- a) Machine language
- b) Low level language
- c) High level language
- d) All of the above

**Ans: a) Machine language**

### **CHAPTER-3: MCO**

---

**Junk e-mail is also called:**

- a) spam
- b) spoof
- c) sniffer script
- d) spool

**Ans: a) spam**

**Office LANs, which are scattered geographically on a large scale, can be connected by the use of corporate:**

- a) CAN
- b) DAN
- c) LAN
- d) WAN

**Ans: d) WAN**



**The device used to carry digital data on analog lines is called as:**

- a) Modem
- b) Multiplexer
- c) Modulator
- d) Demodulator

**Ans: a) Modem**

**Malicious software is known as:**

- a) Badware
- b) Malware
- c) Maliciousware
- d) Illegalware

**Ans: b) Malware**

**A program that performs a useful task while simultaneously allowing destructive acts is:**

- a) Worm
- b) Trojanhorse
- c) Virus
- d) Macrovirus

**Ans: b) Trojanhorse**

**What is the name of an application program that gathers user information and sends it to someone through the Internet ?**

- a) Avirus
- b) Spybot
- c) Logicbomb
- d) Securitypatch

**Ans: b) Spybot**

**ISDN stands for:**

- a) Integrated Services Digital Network
- b) Integrated Subscriber Digital Network
- c) Internet Services Digital Network
- d) Integrated Several Digital Network

**Ans: a) Integrated Services Digital Network**

**Which of the following is a network topology?**

- a) LAN
- b) WAN
- c) MAN
- d) BUS

**Ans: d) BUS**

**The first web browser is:**

- a) Mosaic
- b) Netscape
- c) Internet explorer

d) Collabra

**Ans:a)Mosaic**

**Which one of the following are set of rules and procedures to control the data transmission over the internet:**

a) IP address

b) Domains

c) Protocol

d) Gateway

**Ans:c)Protocol**

#### **CHAPTER-4: MCO**

**Q.1 Which one of the following is collection of related fields that can be treated as a unit by some application program:**

a) field

b) record

c) file

d) database

**Ans:b)record**

**Which of the following is not a part of the usage information?**

a) data created

b) identity of creator

c) owner

d) last date modified

**Ans: c)owner**

**Several instruction executions simultaneously in:**

a) processing

b) parallel processing

c) serial processing

d) multitasking

**Ans:b)parallel processing**

**A term that refers to the way in which the nodes of a network are linked together:**

a) network

b) topology

c) connection

d) interconnectivity

**Ans:b)topology**

**The Details view shows all of the following about a file EXCEPT:**

a) name

b) size

c) type

d) password

**Ans:d)password**

**174|SCTE&VT Learning Materials:Computer Application**

**An easy way to sort files is to:**

a) right click on a file in Details view

- b) click on the column header in Details view
- c) click the sort icon in Details view
- d) alphabetize them

**Ans: b) click on the column header in Details view**

**After creating a file management system on your computer, you should do all of the following**

**EXCEPT:**

- a) delete files that are no longer needed
- b) move files to appropriate folders
- c) rename folders to be more meaningful
- d) run the Task Management

**Ans: a) delete files that are no longer needed**

**When you right-click on a folder on the hard drive and choose Delete, the files:**

- a) are erased
- b) go into the Recycle Bin
- c) are moved into the header section of the hard drive
- d) go into the Old Documents folder

**Ans: b) go into the Recycle Bin**

**One of the first steps when creating a file management system is to:**

- a) create new folders
- b) delete files that will be moved
- c) change to Details view
- d) select multiple files

**Ans: a) create new folders**

**Folder names should:**

- a) use numbers only
- b) be as short as possible
- c) not contain spaces
- d) be meaningful and recognizable

**Ans: d) be meaningful and recognizable**

175 | SCTE & VT Learning Materials: Computer Application

## **CHAPTER-5: MCO**

---

**An algorithm represented in the form of programming language is \_\_\_\_\_**

- a) Flowchart
- b) Pseudocode
- c) Program
- d) None

**Ans: c) Program**

**Keep the statement language \_\_\_\_\_ while writing pseudocode.**

- a) Dependent
- b) Independent

- c) Case sensitive
- d) Capitalized

**Ans:**b)Independent

**Flowcharts and Algorithms are used for**

- a) Better Programming
- b) Easy testing and Debugging
- c) Efficient Coding
- d) All

**Ans:**d) All

**Which of the following is not a keyword?**

- a) Read
- b) Write
- c) Start
- d) Endif

**Ans:**c)Start

**\_\_\_\_\_ is used to show hierarchy in pseudocode.**

- a) Indentation
- b) Curly Braces
- c) Round Brackets
- d) Semicolon

**Ans:**a)Indentation

**\_\_\_\_\_ are identified by their addresses, we give them names (field names /variable names) using words.**

- a) Memory variables
- b) Memory Locations
- c) Memory Addresses
- d) Data variables

**Ans:**b) Memory Locations

**\_\_\_\_\_ begins with lowercase letters.**

- a) Keywords
- b) Variables
- c) Tokens
- d) Functions

**Ans:**b)Variables

**A symbol used for grouping.**

- a) ()
- b) {}
- c) []
- d) —|

**Ans:**a) ()

**A statement used to close the IF block.**

- a) ELSE
- b) ELSEIF
- c) END
- d) ENDIF

**Ans:**d) ENDIF

**Instructurallanguage,wecan'taddanew sortof**

- a) Loop
- b) Function
- c) Variable
- d) Constant

**Ans:**a)Loop

### **CHAPTER-6: MCO**

**Who invented C Language?**

- a) Charles Babbage
- b) Grahambel
- c) Dennis Ritchie
- d) SteveJobs

**Ans:**c) DennisRitchie

**C is \_\_\_\_\_ type of programming language?**

- a) ObjectOriented
- b) Procedural
- c) Bitlevellanguage
- d) Functional

**Ans:**b) Procedural

**A C program is a combination of?**

- a) Statements
- b) Functions
- c) Variables
- d) All of the above

**Ans:**d) All of the above

**Types of Integers are?**

- a) short
- b) int
- c) long
- d) All the above

**Ans:**d) All the above

**Choose a correct statement about C break; statement?**

- a) break; statement can be used inside switch block
- b) break; statement can be used with loops like for, while and do while.
- c) break; statement causes only the same or inner loop where break; is present to quit suddenly.
- d) All the above.

**Ans:**d) All the above.

**What is the output of the C Program?**

```
int main()
{
if(4<5)
printf("Hurray..\n");
printf("Yes");else
printf("England")
```

```
return 0;
```

```
}
```

e) Hurray.. Yes

f) Hurray..

Yes

g) Compiler error

h) None of the above

**Ans:** c) Compiler error

**Operator % in C Language is called?**

a) Percentage Operator

b) Quotient Operator

c) Modulus

d) Division

**Ans:** c) Modulus

**Can you use C Modulo Division operator % with float and int?**

a) Only int variables = Okay

b) Only float variables = Okay

c) int or float combination = Okay

d) Numerator int variable, Denominator any variable = Okay

**Ans:** a) Only int variables = Okay

**Which loop is faster in C Language, for, while or Do While?**

a) for

b) while

c) do while

d) All work at same speed

**Ans:** d) All work at same speed

**What is the output of C program with switch statement or block?**

```
int main()
```

```
{
```

```
int a;
```

```
switch(a);
```

```
{
```

```
printf("DEER");
```

```
}
```

```
printf("LION");
```

```
}
```

a) LION

b) DEERLION

c) Compiler error

d) None of the above

**Ans:** b) DEERLION

## **CHAPTER-7: MCO**

**Q.1 Choose correct statement about Functions in C Language.**

- a) A Function is a group of statements which can be reused any number of times.
- b) Every Function has a return type.
- c) Every Function may or may not return a value.
- d) All the above.

**Ans:** d) All the above.

**Q.2 A function which calls itself is called a \_\_\_\_\_ function.**

- a) Self Function
- b) Auto Function
- c) Recursive Function
- d) Static Function

**Ans:** c) Recursive Function

**Q.3 How many values can a C Function return at a time?**

- a) Only One Value
- b) Maximum of two values
- c) Maximum of three values
- d) Maximum of 8 values

**Ans:** a) Only One Value

**Q.4 What are types of Functions in C Language?**

- a) Library Functions
- b) User Defined Functions
- c) Both Library and User Defined
- d) None of the above

**Ans:** c) Both Library and User Defined

**Q.5 Every C Program should contain which function?**

- a) printf()
- b) show()
- c) scanf()
- d) main()

**Ans:** d) main()

**Q.6 What is the maximum number of statements that can present in a C function?**

- a) 64
- b) 128
- c) 256
- d) None of the above

**Ans:** d) None of the above

**Q.7 Arguments passed to a function in C language are called \_\_\_\_\_ arguments.**

- a) Formal arguments
- b) Actual Arguments
- c) Definite Arguments
- d) Ideal Arguments

**Ans:** b) Actual Arguments

**Q.8 An array Index starts with?**

- a) -1
- b) 0
- c) 1
- d) 2

**Ans:** b) 0

**Q.9 What is the output of C program with arrays and pointers?**

```
int main()
{
    int a[3] = {20, 30, 40};
    int *p[3]; p = &a;
    printf("%d", *p[0]);
}
```

- a) 20
- b) address of element 20
- c) Garbage value
- d) Compiler error

**Ans:** d) Compiler error

**Q.10 What is actually passed to PRINTF or SCANF functions?**

- a) Value of String
- b) Address of String
- c) End address of String
- d) Integer equivalent value of String

**Ans:** b) Address of String