MAA OMWATI DEGREE COLLEGE HASSANPUR

(PALWAL)

NOTES

BCA 1ST Sem

Computer Fundamentals and Problem Solving Using C

Unit-1

Generations of computers:-

- **First Generation of Computers(1940 – 1956):** The period from 1940-1956 was the period of first-generation computers. They are basically based on vacuum tubes, and vacuum tubes are used as the basic components for memory and circuitry for CPU (Central Processing Unit). For e.g., UNIVAC-1 And ENIVAC
- **Second Generation of Computers (1957 – 1963):** The second generation of computers consists of two types of devices transistors and magnetic core. For e.g., IBM 1401, IBM 1920, etc.
- **Third Generation of Computers(1964 – 1971):** Jack Kilby invented the Integrated Circuit. This Integrated Circuit replaced the use of transistors in the third generation of computers. Integrated Circuits itself consists of many transistors, capacitors, and resistors and due to this third-generation computers are smaller in size, efficient, and more reliable. For e.g., CDC 1700, IBM-360 Series, etc.
- **Fourth Generation of Computers(1972 onward):** VLSI (Very Large Scale Integrated) Circuit or they are also known as microprocessors are used in Fourth generation computers. Microprocessor chip is made up of thousands of Integrated Circuits build on a single silicon chip. The use of Personal Computer(PCs) increased in this generation and First Personal Computer (PC) was developed by IBM. For e.g., Apple, CRAY-1, etc.
- **Fifth Generation of Computers(Present and Future):** It is based on Artificial intelligence (AI) software. Artificial intelligence describes the medium and way of making computers like humans, the way human think, the way humans act, etc. and this is an emerging branch and has all the scopes for research work too. For e.g., PARAM 10000, IBM notebooks, etc.

Characteristics of Computer:-

1. Speed

Executing mathematical calculation, a computer works faster and more accurately than human. Computers have the ability to process so many millions (1,000,000) of instructions per second. Computer operations are performed in micro and nano seconds. A computer is a time saving device. It performs several calculations and tasks in few seconds that we take hours to solve. The speed of a computer is measure in terms of GigaHertz and MegaHertz.

2. Diligence

A human cannot work for several hours without resting, yet a computer never tires. A computer can conduct millions of calculations per second with complete precision without stopping. A computer can consistently and accurately do millions of jobs or calculations. There is no weariness or lack of concentration. Its memory ability also places it ahead of humans.

3. Reliability

A computer is reliable. The output results never differ unless the input varies. the output is totally depend on the input. when an input is the same the output will also be the same. A computer produces consistent results for similar sets of data, if we provide the same set of input at any time we will get the same result.

4. Automation

The world is quickly moving toward AI (Artificial Intelligence)-based technology. A computer may conduct tasks automatically after instructions are programmed. By executing jobs automatically, this computer feature replaces thousands of workers. Automation in computing is often achieved by the use of a program, a script, or batch processing.

5. Versatility

Versatility refers to a capacity of computer. Computer perform different types of tasks with the same accuracy and efficiency. A

computer can perform multiple tasks at the same time this is known as versatility. For example, while listening to music, we may develop our project using PowerPoint and Wordpad, or we can design a website.

6. Memory

A computer can store millions of records. these records may be accessed with complete precision. Computer memory storage capacity is measured in Bytes, Kilobytes(KB), Megabytes(MB), Gigabytes(GB), and Terabytes(TB). A computer has built-in memory known as primary memory.

7. Accuracy

When a computer performs a computation or operation, the chances of errors occurring are low. Errors in a computer are caused by human's submitting incorrect data. A computer can do a variety of operations and calculations fast and accurately.

Applications of Computer:-

- **Medical**
  The computer plays a very important role in medical science such as record patients' information monitoring heart rate, oxygen level, and blood pressure. To conduct various surgeries junior doctors get the help of another professional doctor by web conferencing. Research is also spread with the help of computers in the health sector.

- **Education**
  Today learning becomes easy because of computers. Anyone employed or student can learn any stage of life with the help of a computer. Computers are very crucial for online classes,

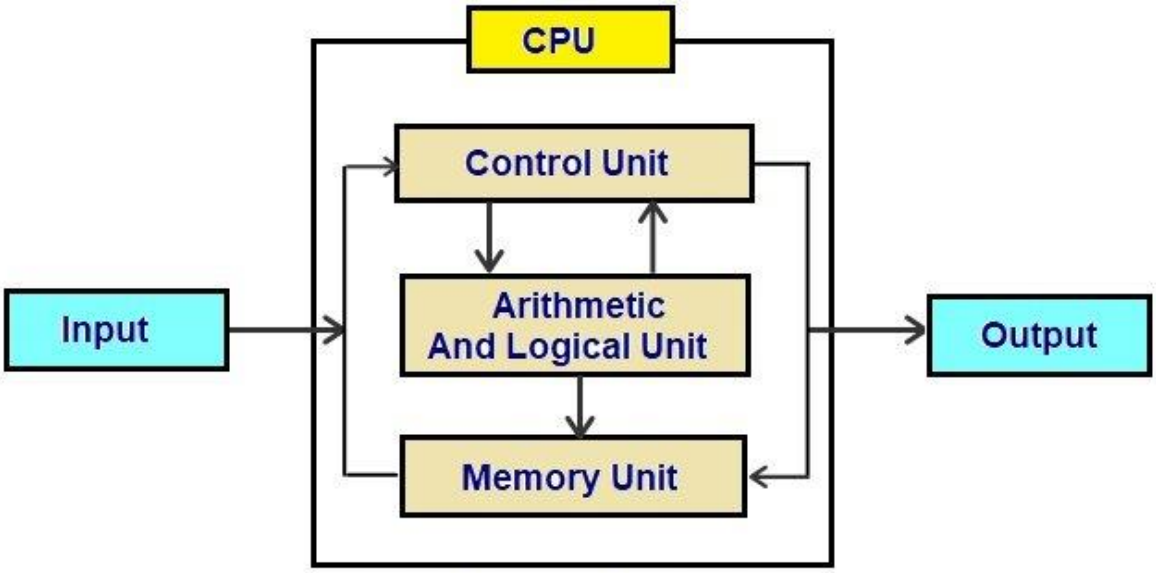
download study material on the internet. Computers are also used to track student attendance and learning strategies. Coaching and institutes increased their areas by audio-visual aids using computers.

- **Banking**
  A computer performs a crucial role in banking sectors, by storing several account holder details on a bank server. All transactions such as deposits and withdrawals perform by a computer. A banking company can easily monitor all ATMs and passbook printing machines.

- **Government Sectors**
  Government can easily monitor government sectors such as road services, railway, development, and other rising funds.

- **Entertainment**
  Today most people are so busy and they do not easily get time to fresh their mind. We can play various interesting video games using a computer. We can watch movies, TV shows, and reality shows on the computer. A computer is also used to create sarcastic memes and make us happy.

Block Diagram along with its components:-



Block Diagram of a Computer

ArtOfTesting

Input

All the data received by the computer goes through the input unit. The input unit comprises different devices like a mouse, keyboard, scanner, etc. In other words, each of these devices acts as a mediator between the users and the computer.

CPU – Central Processing Unit

Central Processing Unit or the CPU, is the brain of the computer. It works the same way a human brain works. As the brain controls all human activities, similarly the CPU controls all the tasks.

Moreover, the CPU conducts all the arithmetical and logical operations in the computer.

Now the CPU comprises of two units, namely – ALU (Arithmetic Logic Unit) and CU (Control Unit). Both of these units work in sync. The CPU processes the data as a whole.

ALU – Arithmetic Logic Unit

The Arithmetic Logic Unit is made of two terms, arithmetic and logic. There are two primary functions that this unit performs.

1. Data is inserted through the input unit into the primary memory. Performs the basic arithmetical operations on it, like addition, subtraction, multiplication, and division. It performs all sorts of calculations required on the data. Then, it sends back data to the storage.
2. The unit is also responsible for performing logical operations like AND, OR, Equal to, Less than, etc. In addition to this, it conducts merging, sorting, and selection of the given data.

CU – Control Unit

The control unit as the name suggests is the controller of all the activities/tasks and operations. All this is performed inside the computer.The memory unit sends a set of instructions to the control unit. Then the control unit in turn converts those instructions. After that these instructions are converted to control signals. These control signals help in prioritizing and scheduling activities. Thus, the control unit coordinates the tasks inside the computer in sync with the input and output units.

Memory Unit

All the data that has to be processed or has been processed is stored in the memory unit. The memory unit acts as a hub of all the data. It transmits it to the required part of the computer whenever necessary.

The memory unit works in sync with the CPU. This helps in faster accessing and processing of the data. Thus, making tasks easier and quicker.

Types of Computer Memory

There are two types of computer memory-

Primary Memory

This type of memory cannot store a vast amount of data. Therefore, it is only used to store recent data. The data stored in this is temporary. It can get erased once the power is switched off. Therefore, is also called temporary memory or main memory.

RAM stands for Random Access Memory. It is an example of primary memory. This memory is directly accessible by the CPU. It is used for reading and writing purposes. For data to be processed, it has to be first transferred to the RAM and then to the CPU.

Secondary Memory

For permanent storage purposes, secondary memory is used. It is also called permanent memory or auxiliary memory. The hard disk is an example of secondary memory. Even in a power failure data does not get erased easily.

Classification of Computers:-

The computer systems can be classified on the following basis:

1. On the basis of size.
2. On the basis of functionality.
3. On the basis of data handling.

Classification on the basis of size:-

1. **Super computers :** The super computers are the most high performing system. A supercomputer is a computer with a high level of performance compared to a general-purpose computer.

2. **Mainframe computers :** These are commonly called as big iron, they are usually used by big organisations for bulk data processing such as statistics, census data processing, transaction processing and are widely used as the servers as these systems has a higher processing capability as compared to the other classes of computers.

3. **Mini computers :** These computers came into the market in mid 1960s and were sold at a much cheaper price than the main frames, they were actually designed for control, instrumentation, human interaction, and communication switching as distinct from calculation and record keeping, later they became very popular for personal uses with evolution.

4. **Micro computers :** A microcomputer is a small, relatively inexpensive computer with a microprocessor as its CPU. It includes a microprocessor, memory, and minimal I/O circuitry mounted on a single printed circuit board.The previous to these computers, mainframes and minicomputers, were comparatively much larger, hard to maintain and more expensive. They actually formed the foundation for present day microcomputers and smart gadgets that we use in day to day life.
   Eg: Tablets, Smartwatches.

Classification on the basis of functionality:-

1. **Servers :** Servers are nothing but dedicated computers which are set-up to offer some services to the clients. They are named depending on the type of service they offered. Eg: security server, database server.
2. **Workstation :** Those are the computers designed to primarily to be used by single user at a time. They run multi-user operating systems. They are the ones which we use for our day to day personal / commercial work.
3. **Information Appliances :** They are the portable devices which are designed to perform a limited set of tasks like basic calculations, playing multimedia, browsing internet etc. They are generally referred as the mobile devices. They have very limited memory and flexibility and generally run on “as-is” basis.
4. **Embedded computers :** They are the computing devices which are used in other machines to serve limited set of requirements. They follow instructions from the non-volatile memory and they are not required to execute reboot or reset. The processing units used in such device work to those basic requirements only and are different from the ones that are used in personal computers-better known as workstations.

Classification on the basis of data handling:-

1. **Analog :** An analog computer is a form of computer that uses the continuously-changeable aspects of physical fact such as electrical, mechanical, or hydraulic quantities to model the problem being solved. Any thing that is variable with respect to time and continuous can be claimed as analog just like an analog clock measures time by means of the distance traveled for the spokes of the clock around the circular dial.

2. **Digital :** A computer that performs calculations and logical operations with quantities represented as digits, usually in the binary number system of “0” and “1”, “Computer capable of solving problems by processing information expressed in discrete form. from manipulation of the combinations of the binary digits, it can perform mathematical calculations, organize and analyze data, control industrial and other processes, and simulate dynamic systems such as global weather patterns.

3. **Hybrid :** A computer that processes both analog and digital data, Hybrid computer is a digital computer that accepts analog signals, converts them to digital and processes them in digital form.

Cache Memory:-

Cache memory is a small, high-speed storage area in a computer. The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations. There are various independent caches in a CPU, which store instructions and data. The most important use of cache memory is that it is used to reduce the average time to access data from the main memory.

By storing this information closer to the CPU, cache memory helps speed up the overall processing time. Cache memory is much faster than the main memory (RAM). When the CPU needs data, it first checks the cache. If the data is there, the CPU can access it quickly. If not, it must fetch the data from the slower main memory.

Computer Network:-

- **Computer Network** is a group of computers connected with each other through wires, optical fibres or optical links so that various devices can interact with each other through a network.
- The aim of the computer network is the sharing of resources among various devices.
- In the case of computer network technology, there are several types of networks that vary from simple to complex level.

Types of Computer Network:-

Personal Area Network (PAN)

PAN is the most basic type of computer network. It is a type of network designed to connect devices within a short range, typically around one person. It allows your personal devices, like smartphones, tablets, laptops, and wearables, to communicate and share data with each other. PAN offers a network range of 1 to 100 meters from person to device providing communication. Its transmission speed is very high with very easy maintenance and very low cost. This uses Bluetooth as technology. Examples of PAN are USB, computer, phone, tablet, printer, PDA, etc.

Local Area Network (LAN)

LAN is the most frequently used network. A LAN is a computer network that connects computers through a common communication path, contained within a limited area, that is, locally. A LAN encompasses two or more computers connected over a server. The two important technologies involved in this network are Ethernet and Wi-fi. It ranges up to 2km & transmission speed is very high with easy maintenance and low cost. Examples of LAN are networking in a home, school, library, laboratory, college, office, etc.

Metropolitan Area Network (MAN)

A MAN is larger than a LAN but smaller than a WAN. This is the type of computer network that connects computers over a geographical distance through a shared communication path over a city, town, or metropolitan area. This network mainly uses FDDI, CDDI, and ATM as the technology with a range from 5km to 50km. Its transmission speed is average. It is difficult to maintain and it comes with a high cost. Examples of MAN are networking in towns, cities, a single large city, a large area within multiple buildings, etc.

Wide Area Network (WAN)

WAN is a type of computer network that connects computers over a large geographical distance through a shared communication path. It is not restrained to a single location but extends over many locations. WAN can also be defined as a group of local area networks that communicate with each other with a range above 50km. Here we use Leased-Line & Dial-up technology. Its transmission speed is very low and it comes with very high maintenance and very high cost. The most common example of WAN is the Internet.

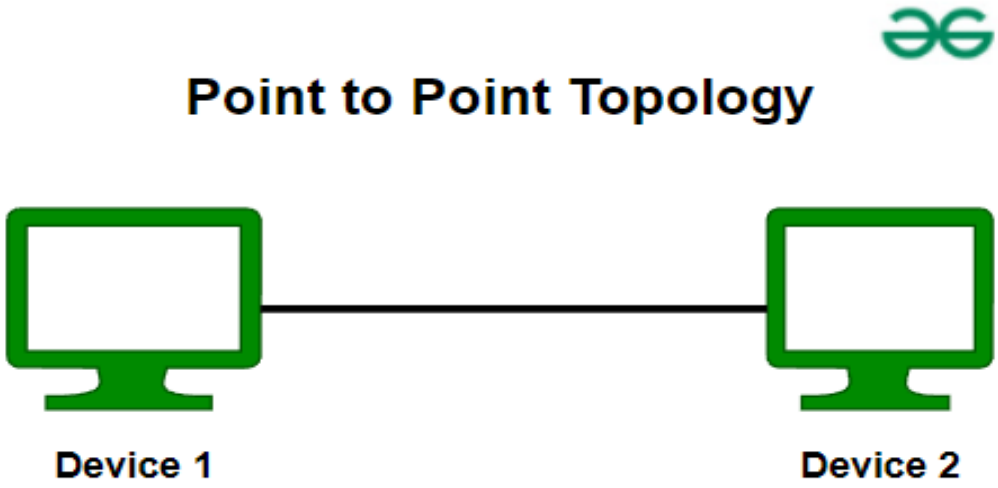
Network Topology:-

Network topology refers to the arrangement of different elements like nodes, links, or devices in a computer network. It defines how these components are connected and interact with each other.

Types of Network Topology:-

Point to Point Topology

Point-to-point topology is a type of topology that works on the functionality of the sender and receiver. It is the simplest communication between two nodes, in which one is the sender and the other one is the receiver. Point-to-Point provides high bandwidth.



Mesh Topology

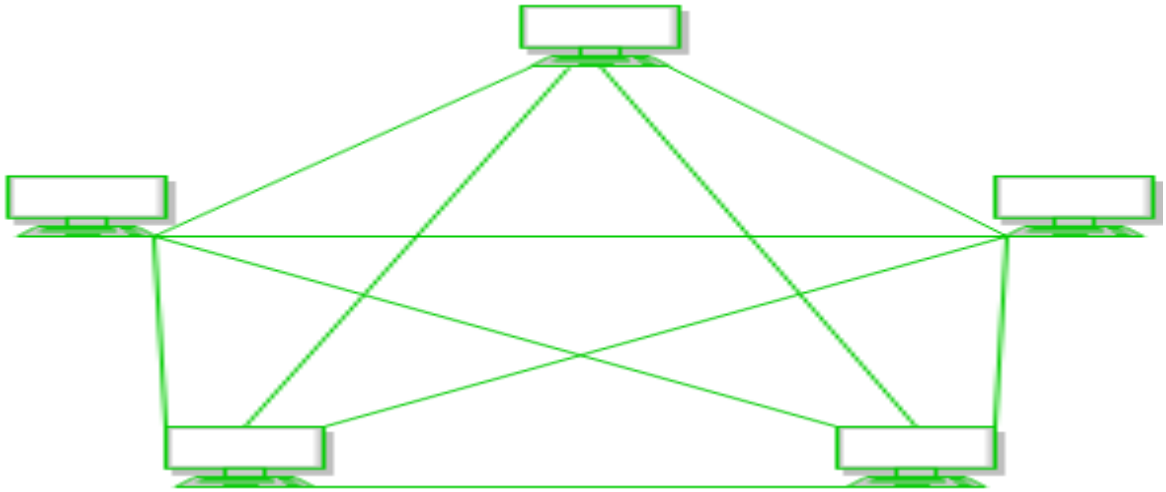
In a mesh topology, every device is connected to another device via a particular channel. In Mesh Topology, the protocols used are AHCP (Ad Hoc Configuration Protocols), DHCP (Dynamic Host Configuration Protocol), etc.

Advantages of Mesh Topology

- Communication is very fast between the nodes.
- Mesh Topology is robust.
- The fault is diagnosed easily. Data is reliable because data is transferred among the devices through dedicated channels or links.
- Provides security and privacy.

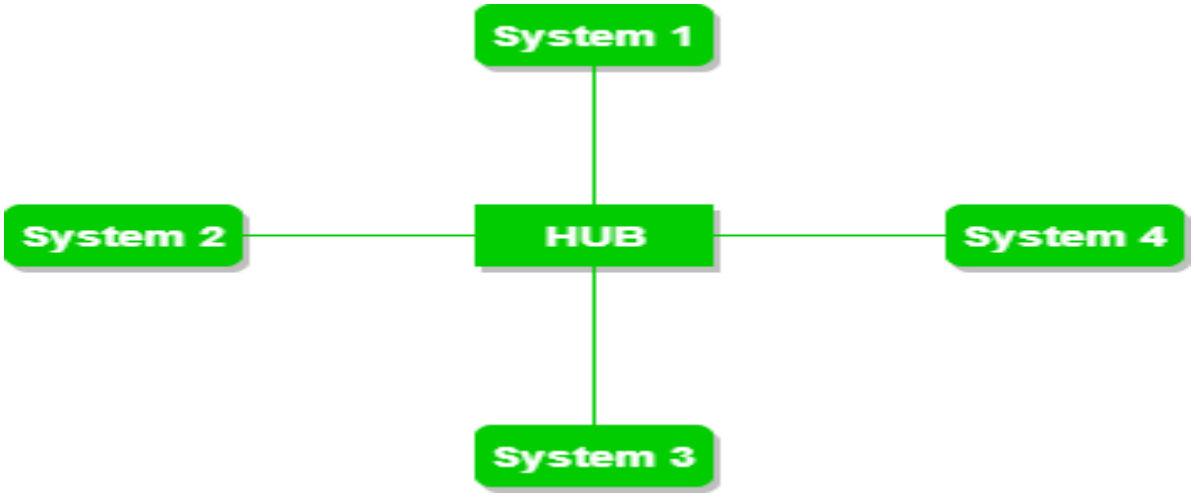
Disadvantages of Mesh Topology

- Installation and configuration are difficult.
- The cost of cables is high as bulk wiring is required, hence suitable for less number of devices.
- The cost of maintenance is high.



Star Topology

In Star Topology, all the devices are connected to a single hub through a cable. This hub is the central node and all other nodes are connected to the central node. The hub can be passive in nature i.e., not an intelligent hub such as broadcasting devices, at the same time the hub can be intelligent known as an active hub. Active hubs have repeaters in them. Coaxial cables or RJ-45 cables are used to connect the computers. In Star Topology, many popular Ethernet LAN protocols are used as CD(Collision Detection), CSMA (Carrier Sense Multiple Access), etc.



Advantages of Star Topology

- It is Robust. If one link fails only that link will affect and not other than that.
- Easy to fault identification and fault isolation.
- Star topology is cost-effective as it uses inexpensive coaxial cable.

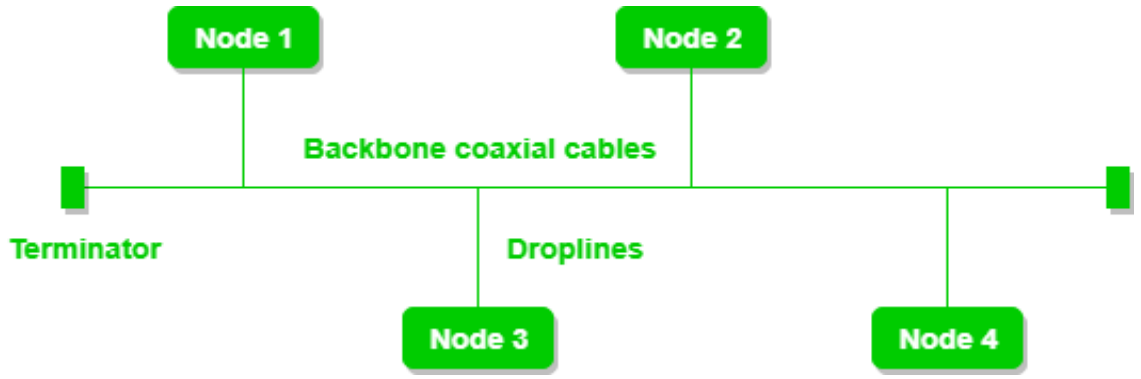
Disadvantages of Star Topology

- If the concentrator (hub) on which the whole topology relies fails, the whole system will crash down.
- The cost of installation is high.
- Performance is based on the single concentrator i.e. hub.

Bus Topology

Bus Topology is a network type in which every computer and network device is connected to a single cable. It is bi-directional. It is a multi-point connection and a non-robust topology because if the backbone fails the topology crashes. In Bus Topology, various MAC (Media

Access Control) protocols are followed by LAN ethernet connections like TDMA , Pure Aloha , CDMA, Slotted Aloha , etc.



Advantages of Bus Topology

- Coaxial or twisted pair cables are mainly used in bus-based networks that support up to 10 Mbps.
- The cost of the cable is less compared to other topologies, but it is used to build small networks.
- Bus topology is familiar technology as installation and troubleshooting techniques are well known.
- CSMA is the most common method for this type of topology.
-

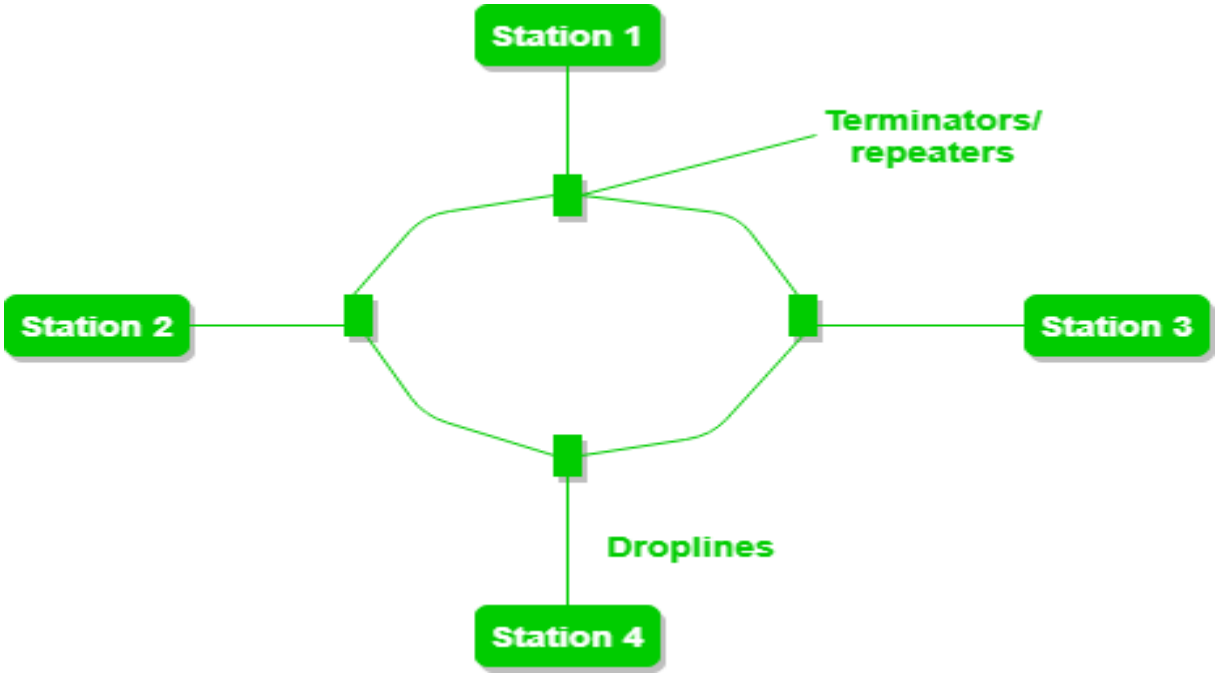
Disadvantages of Bus Topology

- A bus topology is quite simpler, but still, it requires a lot of cabling.
- If the common cable fails, then the whole system will crash down.
- If the network traffic is heavy, it increases collisions in the network. To avoid this, various protocols are used in the MAC layer known as Pure Aloha, Slotted Aloha, CSMA/CD, etc.

- Adding new devices to the network would slow down networks.
- Security is very low.

Ring Topology

In a Ring Topology, it forms a ring connecting devices with exactly two neighboring devices.The data flows in one direction, i.e. it is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called Dual Ring Topology. In-Ring Topology, the Token Ring Passing protocol is used by the workstations to transmit the data.



The most common access method of ring topology is token passing.

- **Token passing:** It is a network access method in which a token is passed from one node to another node.
- **Token:** It is a frame that circulates around the network.

Advantages of Ring Topology

- The data transmission is high-speed.
- The possibility of collision is minimum in this type of topology.
- Cheap to install and expand.
- It is less costly than a star topology.

Disadvantages of Ring Topology

- The failure of a single node in the network can cause the entire network to fail.
- Troubleshooting is difficult in this topology.
- The addition of stations in between or the removal of stations can disturb the whole topology.
- Less secure.

Tree Topology

This topology is the variation of the Star topology. This topology has a hierarchical flow of data. In Tree Topology, protocols like DHCP and SAC (Standard Automatic Configuration) are used.

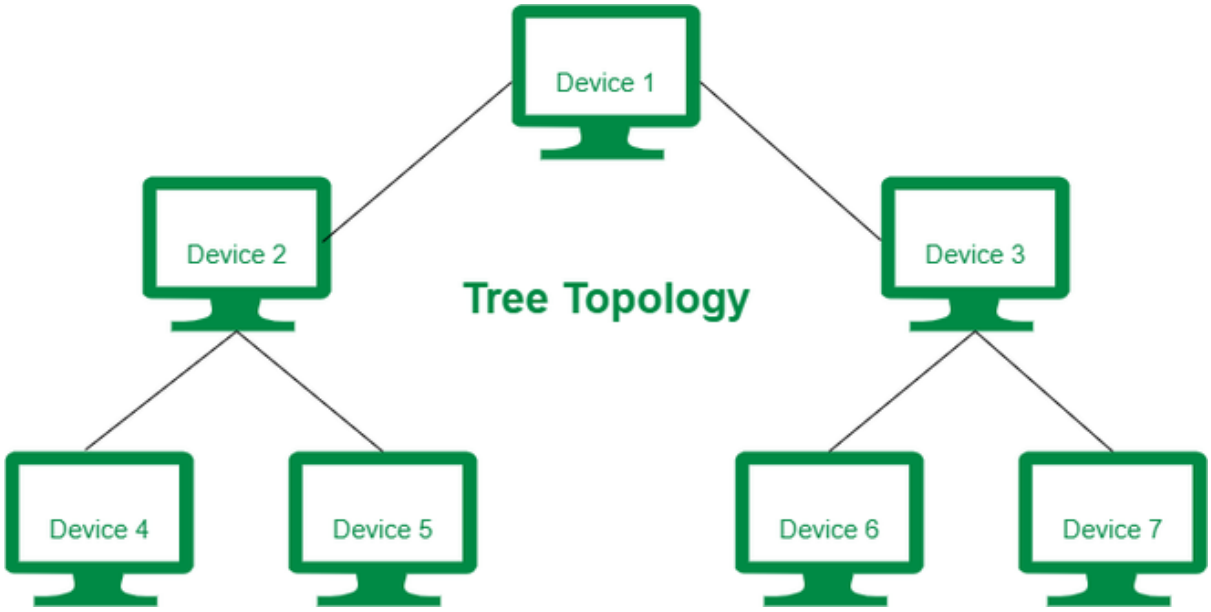
Advantages of Tree Topology

- It allows more devices to be attached to a single central hub thus it decreases the distance that is traveled by the signal to come to the devices.
- It allows the network to get isolated and also prioritize from different computers.
- We can add **new devices to the existing network.**

- **Error detection** and **error correction** are very easy in a tree topology.

Disadvantages of Tree Topology

- If the central hub gets fails the entire system fails.
- The cost is high because of the cabling.
- If new devices are added, it becomes difficult to reconfigure.



Hybrid Topology

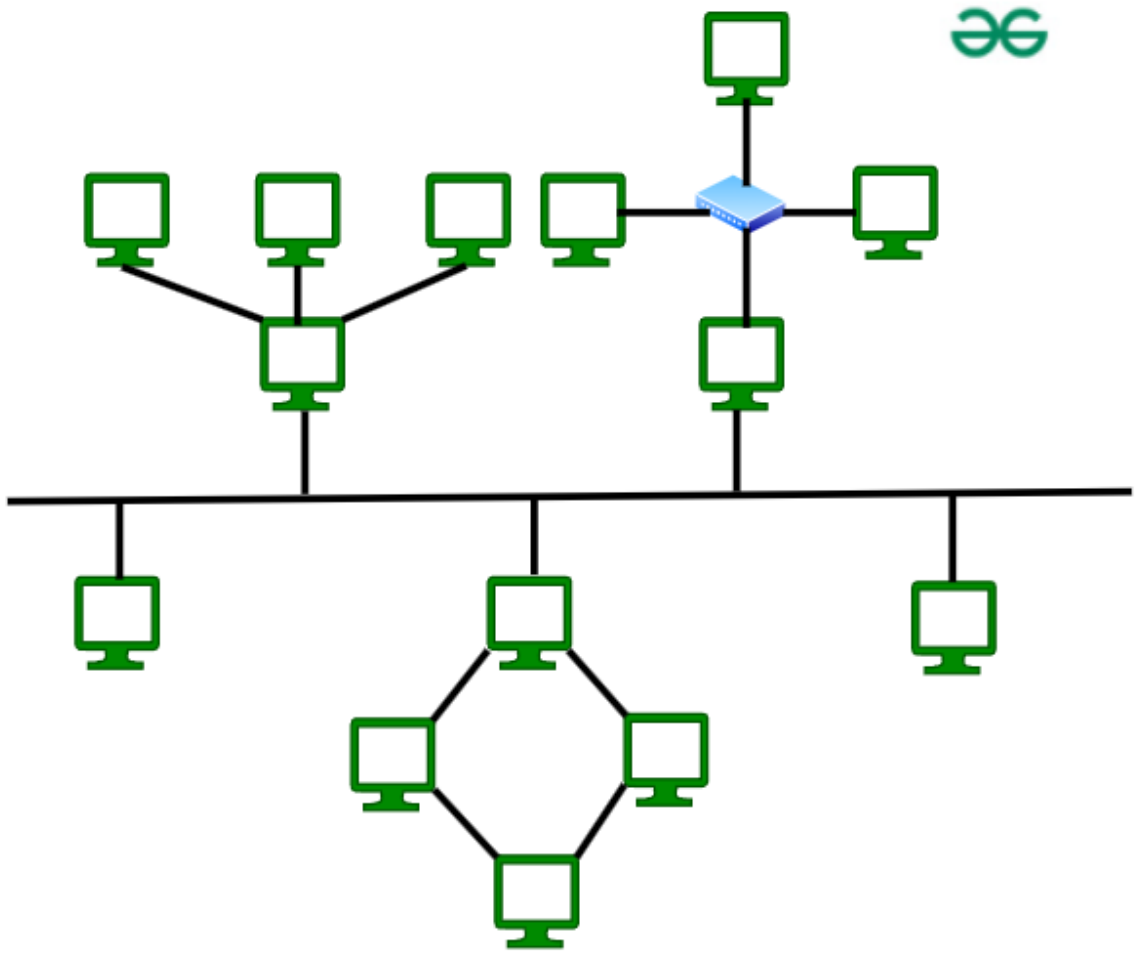
This topological technology is the combination of all the various types of topologies we have studied above. Hybrid Topology is used when the nodes are free to take any form. It means these can be individuals such as Ring or Star topology or can be a combination of various types of topologies.

Advantages of Hybrid Topology

- This topology is **very flexible** .
- The size of the network can be easily expanded by **adding new devices.**

Disadvantages of Hybrid Topology

- It is challenging **to design the architecture** of the Hybrid Network.
- **Hubs** used in this topology are **very expensive.**
- The infrastructure cost is very high as a hybrid network **requires a lot of cabling and network devices.**



Internet and Its Applications:-

The Internet is the foremost important tool and the prominent resource that is being used by almost every person across the globe. It connects millions of computers, webpages, websites, and servers. Using the internet we can send emails, photos, videos, and messages to our loved ones. Or in other words, the Internet is a widespread interconnected network of computers and electronic devices(that support Internet). It creates a communication medium to share and get information online. If your device is connected to the Internet then only you will be able to access all the applications, websites, social media apps, and many more services. The Internet nowadays is considered the fastest medium for sending and receiving information.

Applications:-

- **Online Banking and Transaction:** The Internet allows us to transfer money online through the net banking system. Money can be credited or debited from one account to the other.
- **Education, Online Jobs, Freelancing:** Through the Internet, we are able to get more jobs via online platforms like Linkedin and to reach more job providers. Freelancing on the other hand has helped the youth to earn a side income and the best part is all this can be done via the INTERNET.

- **Entertainment:** There are numerous options for entertainment online we can listen to music, play games can watch movies, and web series, and listen to podcasts, youtube itself is a hub of knowledge as well as entertainment.
- **New Job Roles:** The Internet has given us access to social media, and digital products so we are having numerous new job opportunities like digital marketing and social media marketing online businesses are earning huge amounts of money just because the Internet is the medium to help us to do so.
- **Best Communication Medium:** The communication barrier has been removed from the Internet. You can send messages via email, Whatsapp, and Facebook. Voice chatting and video conferencing are also available to help you to do important meetings online.
- **Comfort to humans:** Without putting any physical effort you can do so many things like shopping online it can be anything from stationeries to clothes, books to personal items, etc. You can books train and plane tickets online.
- **GPS Tracking and google maps:** Yet another advantage of the internet is that you are able to find any road in any direction, and areas with less traffic with the help of GPS on your mobile.

Operating System

An operating system is a program that manages a computer's hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and computer hardware. The main task an operating system carries out is the allocation of resources and services, such as the allocation of memory, devices, processors, and information. The operating system also includes programs to manage these resources, such as a traffic controller, a scheduler, a memory management module, I/O programs, and a file system. The operating system simply provides an environment within which other programs can do useful work.

Functions:-

An Operating System acts as a communication bridge (interface) between the user and computer hardware. The purpose of an operating system is to provide a platform on which a user can execute programs conveniently and efficiently.

An operating system is a piece of software that manages the allocation of Computer Hardware. The coordination of the hardware must be appropriate to ensure the correct working of the computer system and to prevent user programs from interfering with the proper working of the system.

The main goal of the Operating System is to make the computer environment more convenient to use and the Secondary goal is to use the resources most efficiently.

Unit-2

Problem Definition

Definition: Problem definition involves identifying and clearly stating the problem that needs to be solved by a program.
Key steps include:

1. Understanding the requirements (input, processing, output).
2. Breaking the problem into smaller, manageable parts.
3. Identifying constraints and edge cases.

Program Design

Program design is the process of planning a solution before coding. It involves:

1. **Defining Inputs and Outputs**:
    - What data will the program accept?
    - What results should it produce?
2. **Developing a Logical Flow**:
    - Identify steps to achieve the solution.
    - Structure these steps in a clear and logical order.
3. **Choosing a Programming Approach**:
    - Procedural or object-oriented, based on the problem.
4. **Representing the Plan**:
    - Use tools like algorithms, pseudocode, and flowcharts.

Debugging

Definition: Debugging is the process of identifying, analyzing, and removing errors in a program.

Steps in Debugging:

1. **Reproduce the Error**: Understand when and how the error occurs.

2. **Locate the Error**: Use debugging tools, logging, or step-by-step code analysis.
3. **Fix the Error**: Correct the faulty logic or syntax.
4. **Test the Fix**: Ensure that the error is resolved without introducing new ones.

Types of Errors in Programming

1. **Syntax Errors**:
    - Occur due to violations of language rules.
    - Example: Missing a semicolon or unmatched parentheses.
2. **Logical Errors**:
    - The program runs but produces incorrect results.
    - Example: Incorrect formula implementation.
3. **Runtime Errors**:
    - Occur during program execution.
    - Example: Division by zero or accessing an out-of-bounds array.
4. **Compilation Errors**:
    - Errors that prevent the program from being compiled.
    - Example: Incorrect function signatures.

Techniques of Problem Solving

1. Flowcharting

A **flowchart** is a graphical representation of a program's logic or process flow.
Symbols:

- **Oval**: Start/End.
- **Rectangle**: Process or operation.
- **Diamond**: Decision point.
- **Arrow**: Flow of control.

Advantages:

- Visual clarity of processes.
- Useful for debugging and understanding logic.

2. Algorithms

An **algorithm** is a step-by-step procedure to solve a problem.
Characteristics:

- Finiteness: Must terminate after a finite number of steps.
- Definiteness: Steps should be clear and unambiguous.
- Input/Output: Accepts inputs and produces outputs.
- Effectiveness: Each step must be basic enough to be executed.

Example:
Algorithm for Adding Two Numbers:

1. Start.
2. Input two numbers: A and B.
3. Calculate the sum: SUM = A + B.
4. Output the result: SUM.
5. End.

Overview of C Programming Language

History of C

- **Developed by**: Dennis Ritchie in 1972 at Bell Labs.
- **Evolution**:
    - Evolved from languages like **B** and **BCPL**.
    - Initially designed for system programming, particularly for developing the UNIX operating system.
- **Standardization**:

- Standardized by ANSI (American National Standards Institute) in 1989 (ANSI C).
- Further refined by ISO standards (ISO C).

Importance of C

1. **Foundation Language**: Serves as a base for many modern programming languages like C++, Java, Python, and more.
2. **Portability**: Programs written in C are highly portable across platforms.
3. **Performance**: Known for its efficiency and close-to-hardware functionality.
4. **Versatility**: Suitable for developing system software (e.g., operating systems) and application software (e.g., games).
5. **Rich Library Support**: Provides a wide range of built-in functions.

Elements of C

1. Character Set

The basic building blocks of C programs:

- **Letters**: A-Z, a-z.
- **Digits**: 0-9.
- **Special Characters**: +, -, *, /, %, @, #, etc.
- **White Spaces**: Space, tab, newline.
-

2. Identifiers and Keywords

- **Identifiers**:
  Names used for variables, functions, arrays, etc.
  - Must begin with a letter or an underscore (_).
  - Cannot use reserved keywords.

- **Keywords**:
  Predefined reserved words with specific meanings in C. Examples:
    - int, float, if, while, return, etc.

3. Data Types

C provides various data types to define variables:

1. **Basic Types**:
    - int, float, char, double.
2. **Derived Types**:
    - Arrays, pointers, functions.
3. **Enumeration Types**:
    - enum.
4. **Void**:
    - Represents "no type". Used for functions that do not return a value.

4. Constants and Variables

- **Constants**: Fixed values that do not change during program execution. Examples:
    - const int x = 5;
    - #define PI 3.14 (Symbolic Constant).
- **Variables**: Memory locations that store values and can change during execution.
    - Declared using a data type, e.g., int num;.

5. Assignment Statement

Used to assign values to variables. Example:

int a;

a = 10;

6. Symbolic Constants

Symbolic constants are defined using #define directive. Example:

#define PI 3.14159

Structure of a C Program

A C program typically has the following structure:

```
#include <stdio.h>  // Preprocessor directive

int main() {        // Main function
   // Variable declaration
   int a = 10, b = 20, sum;

   // Process
   sum = a + b;

   // Output
   printf("Sum: %d", sum);

   return 0;       // Return statement
}
```

I/O Functions: printf() and scanf()

- **printf()**: Outputs data to the screen. Example:

printf("Hello, World!");

printf("Sum = %d", sum);

- **scanf()**: Takes input from the user. Example:

int x;

scanf("%d", &x); // Reads an integer from the user

Operators and Expressions

Operators in C:

1. **Arithmetic Operators**: +, -, *, /, %.
2. **Relational Operators**: ==, !=, <, >, <=, >=.
3. **Logical Operators**: &&, ||, !.
4. **Bitwise Operators**: &, |, ^, ~, <<, >>.
5. **Assignment Operators**: =, +=, -=, *=, /=, %=.
6. **Unary Operators**: ++, --, sizeof, & (address-of).
7. **Ternary Operator**: ?:.

Expressions:

Combinations of operators and operands. Example:

int result = (a + b) * c;

Type Casting and Conversion

- **Type Casting**: Explicit conversion of one data type to another. Example:

float result = (float) a / b;

- **Type Conversion**: Implicit conversion performed by the compiler. Example:

float x = 5; // Automatically converted to 5.0

Operator Hierarchy and Associativity

- **Operator Precedence**: Determines the order in which operators are evaluated.
  Example:
  * and / have higher precedence than + and -.
- **Associativity**: Determines the direction of evaluation when operators have the same precedence.
  Example:
  - Left to Right: *, /, %.
  - Right to Left: Assignment operators (=).

Example:

```
int result = 10 + 5 * 2;  // Multiplication is evaluated first
```

Unit-3

Decision-Making and Looping in C Programming

C provides various control structures for decision-making, enabling programs to choose between different actions based on conditions.

1. if Statement

Executes a block of code if the condition is true.
Syntax:

```
if (condition) {
   // Code to execute if the condition is true
}
```

Example:

```
int x = 10;
if (x > 0) {
   printf("x is positive.");
}
```

2. if-else Statement

Adds an alternative block to execute if the condition is false.
Syntax:

```
if (condition) {
   // Code to execute if the condition is true
} else {
   // Code to execute if the condition is false
}
```

Example:

```
int x = -5;
if (x > 0) {
   printf("x is positive.");
} else {
   printf("x is not positive.");
}
```

3. Nested if Statement

Allows multiple levels of decision-making by placing one if inside another.

Syntax:

c

Copy code

```
if (condition1) {
   if (condition2) {
      // Code to execute if both conditions are true
   }
}
```

Example:

```
int x = 5, y = 10;
if (x > 0) {
   if (y > 0) {
      printf("Both x and y are positive.");
   }
}
```

4. else-if Ladder

Used when there are multiple conditions to check.

Syntax:

c

Copy code

```
if (condition1) {
   // Code for condition1
} else if (condition2) {
   // Code for condition2
} else {
   // Code if none of the conditions are true
}
```

Example:

```
int x = 0;
if (x > 0) {
   printf("x is positive.");
} else if (x < 0) {
   printf("x is negative.");
} else {
   printf("x is zero.");
}
```

5. switch Statement

Checks a variable against multiple constant values and executes a corresponding block.

Syntax:

```
switch (expression) {
    case value1:
        // Code for value1
        break;
    case value2:
        // Code for value2
        break;
    default:
        // Code if no case matches
}
```

Example:

```
int day = 3;
switch (day) {
    case 1: printf("Monday"); break;
    case 2: printf("Tuesday"); break;
    case 3: printf("Wednesday"); break;
    default: printf("Invalid day");
}
```

6. goto Statement

Transfers control to a labeled statement.

Syntax:

```
goto label;
...
label:
// Code here
```

Example:

```
int x = 10;
if (x > 0) {
   goto positive;
}
printf("x is not positive.");
positive:
printf("x is positive.");
```

Looping in C

Loops execute a block of code multiple times based on a condition.

1. while Loop

Executes as long as the condition is true.
Syntax:

```
while (condition) {
   // Code to execute
}
```

Example:

c

Copy code

```
int i = 1;
while (i <= 5) {
   printf("%d ", i);
   i++;
}
```

2. do-while Loop

Executes the code at least once before checking the condition.
Syntax:

```
do {
   // Code to execute
} while (condition);
```

Example:

```
int i = 1;
do {
   printf("%d ", i);
   i++;
} while (i <= 5);
```

3. for Loop

Used for a known number of iterations.
Syntax:

```
for (initialization; condition; increment/decrement) {
   // Code to execute
}
```

Example:

```
for (int i = 1; i <= 5; i++) {
   printf("%d ", i);
}
```

Jumps in Loops

1. break Statement

Exits a loop immediately when executed.
Example:

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        break;
    }
    printf("%d ", i);  // Output: 1 2
}
```

2. continue Statement

Skips the current iteration and proceeds to the next iteration. **Example**:

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        continue;
    }
    printf("%d ", i);  // Output: 1 2 4 5
}
```

Nested Loops

A loop inside another loop. **Example**:

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 2; j++) {
        printf("i = %d, j = %d\n", i, j);
    }
}
```

Output:

i = 1, j = 1

i = 1, j = 2

i = 2, j = 1

i = 2, j = 2

i = 3, j = 1

i = 3, j = 2

Unit-4

Functions, Arrays, Strings, and Pointers in C Programming

Functions in C

Functions are reusable blocks of code designed to perform a specific task. They help organize and modularize programs.

1. Standard Mathematical Functions

C includes several built-in mathematical functions available in the <math.h> library:

- **sqrt(x)**: Returns the square root of x.
- **pow(x, y)**: Returns x raised to the power y.
- **abs(x)**: Returns the absolute value of x.
- **sin(x), cos(x), tan(x)**: Return trigonometric values.
- **log(x)**: Returns the natural logarithm of x.
- **Example**:

```
#include <stdio.h>
#include <math.h>

int main() {
   double x = 9.0;
   printf("Square root: %.2f\n", sqrt(x));
   return 0;
}
```

2. Input/Output in C

Unformatted I/O Functions

- **getchar()**: Reads a single character from standard input.

- **putchar()**: Writes a single character to standard output.
- **Example**:

```
char c = getchar();
putchar(c);
```

Formatted I/O Functions

- **scanf()**: Reads formatted input.
- **printf()**: Writes formatted output.
- **Example**:

```
int num;
scanf("%d", &num);   // Input an integer
printf("Number: %d", num);
```

3. String Manipulation Functions

Available in <string.h>:

- **strcpy(dest, src)**: Copies string src to dest.
- **strlen(str)**: Returns the length of str.
- **strcmp(str1, str2)**: Compares two strings.
- **strcat(dest, src)**: Concatenates src to the end of dest.
- **Example**:

```
#include <stdio.h>
#include <string.h>

int main() {
   char str1[20] = "Hello";
   char str2[] = " World";
   strcat(str1, str2);
```

```
    printf("%s", str1);  // Output: Hello World
    return 0;
}
```

4. User-Defined Functions

Definition

Functions created by the programmer to perform specific tasks.

Function Prototype

Declares a function's name, return type, and parameters. **Example**:

```
int add(int, int);  // Function prototype
```

Local and Global Variables

- **Local Variables**: Declared inside a function; scope is limited to that function.
- **Global Variables**: Declared outside all functions; accessible throughout the program.

Passing Parameters

1. **Pass by Value**: Copies the value of the argument.
2. **Pass by Reference**: Uses pointers to allow modification of the actual variable.

Recursion

A function calls itself directly or indirectly. **Example**:

```
int factorial(int n) {
    if (n == 0) return 1;
    return n * factorial(n - 1);
}
```

Arrays in C

Arrays store multiple values of the same type.

1. Definition and Types

- **1D Array**: Stores a list of elements.
- **2D Array**: Stores a matrix of elements.
- **Multidimensional Arrays**: Arrays with more than two dimensions.

2. Initialization and Processing

- **Declaration and Initialization**:

```
int arr[5] = {1, 2, 3, 4, 5};
```

- **Processing**:

```
for (int i = 0; i < 5; i++) {
   printf("%d ", arr[i]);
}
```

3. Passing Arrays to Functions

Arrays are passed by reference.
Example:

```
void display(int arr[], int size) {
   for (int i = 0; i < size; i++) {
      printf("%d ", arr[i]);
   }
}
```

Strings in C

Strings are arrays of characters terminated by a null character (\0).

1. Declaration and Initialization

- **Declaration**:

```
char str[10];
```

- **Initialization**:

```
char str[] = "Hello";
```

2. Input/Output of String Data

- **Input**:

```
scanf("%s", str);
```

- **Output**:

```
printf("%s", str);
```

Pointers in C

1. Definition

Pointers store the address of another variable.

2. Declaration and Initialization

- **Declaration**:

```
int *ptr;
```

- **Initialization**:

```
int x = 10;
int *ptr = &x;  // Pointer stores the address of x
```

3. Accessing Values

- **Dereferencing**:

```
printf("%d", *ptr);  // Accesses the value of x via ptr
```

4. Pointers and Arrays

- The name of an array acts as a pointer to its first element. **Example**:

```
int arr[5] = {1, 2, 3, 4, 5};
int *ptr = arr;
printf("%d", *(ptr + 2));  // Outputs 3
```