

PLUS ONE COMPUTER SCIENCE



**STUDY
MATERIAL**



COMPILED BY:

LAVAN I K

HSST COMPUTER SCIENCE
GHSS THANNIAM
THRISSUR



HIGHER SECONDARY NATIONAL SERVICE SCHEME

Chapter – 1 The Discipline of Computing

Evolution of the positional number system:

In 3000BC, Egyptian number system emerged. Its base was 10 and write from right to left.

Later, Sumerian / Babylonian number system developed. Its base was 60 and known as sexagesimal system. Numerals were written from left to right. It was the largest base number system. They did not use symbols for zero.

Chinese number system emerged around in 2500BC. It was the simplest and most efficient number system. It had the base 10, very similar to decimal number system.

Greek number system evolved in 500BC and was known as Ionian number system. It was a decimal number system and they did not have any symbol for zero.

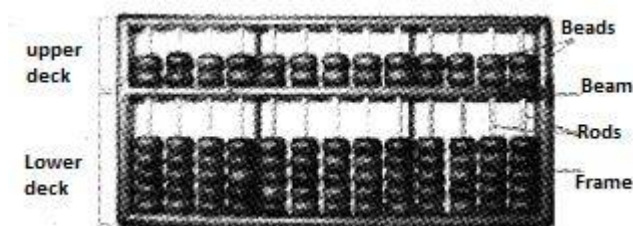
The Romans developed a number system which contains only 7 letters (I, V, X, L, C, D and M). They used mathematics for practical purposes like construction of roads, bridges etc.

The Mayans used a number system with base 20. The number system could produce very accurate astronomical observations and measurements.

Evolution of the computing machine:

(a) Abacus:

In 3000 BC Mesopotamians introduced this and it means calculating board or frame. It is considered as the first computer for basic arithmetical calculations and consists of beads on movable rods divided into two parts. The Chinese improved the Abacus with seven beads on each wire.



(b) Napier's bones:John Napier devised a set of numbering rods known as Napier's bones in 1617 AD. Using it, multiplication problem could be easily done.

(c) Pascaline:Blaise Pascal was French mathematician who developed a computing machine that was capable of adding and subtracting two numbers directly. It could find multiplication and division by repetition. He called it as Pascaline.

(d) Leibniz's calculator: German Mathematician Gottfried Wilhelm von Leibniz designed a calculating machine called the Step Reckoner. Step Reckoner expanded Pascal's idea and successfully introduced it into market.

(e) Jacquard's loom:The Jacquard loom is a mechanical loom invented by Joseph Marie Jacquard in 1801. The loom is controlled by punched cards with punched holes. The loom stored patterns on cards to be utilised again to create the same product.

(f) Difference engine:The intervention of human beings was eliminated by Charles Babbage in calculations by using Difference engine in 1822. It could perform arithmetic operations and print results automatically

(g) Analytical engine:In 1833, Charles Babbage introduced Analytical Engine, the real predecessor of modern computer. It had a store (memory), mill (processor) and input output units. This engine was controlled by programs stored in punched cards. These programs were written by Babbage's assistant, Augusta Ada King, who was considered as the first programmer in the World.

Generations of computers:

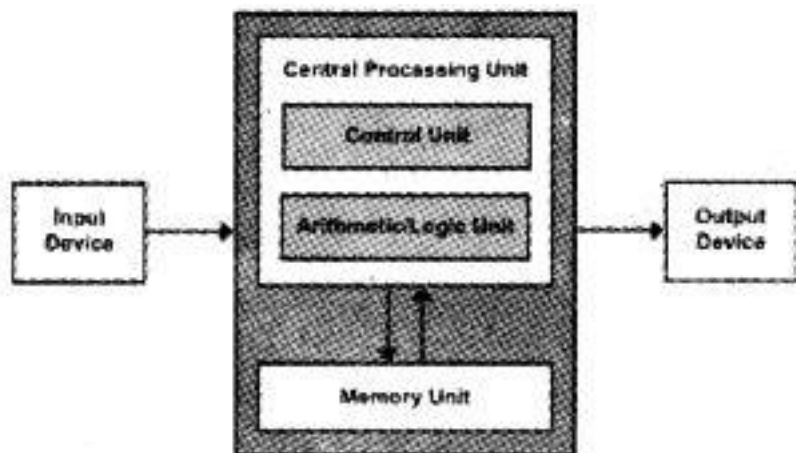
There are five generations of computers from 16th century to till date.

Generations of Computers	Period	Component / Technology
First	1940-1956	Vacuum tubes
Second	1956-1963	Transistors
Third	1964-1971	Integrated Circuits (IC)
Fourth	1971-Present	Microprocessors
Fifth	Present and beyond	Artificial Intelligence

First generation computers (1940 – 1956):

Vacuum tubes were used in first generation computers. The input was based on punched cards and paper tapes and output was displayed on printouts. The Electronic Numerical Integrator and Calculator (ENIAC) belongs to first generation.

Von Neumann architecture



The mathematician John Von Neumann designed a computer structure that structure is in use nowadays. Von Neumann structure consists of a central processing unit (CPU), Memory unit, Input and Output unit. The CPU consists of arithmetic logical unit (ALU) and control unit (CU). The instructions are stored in the memory and follows the “Stored Program Concept”.

Second generation computers (1956 -1963):

Transistors, instead of vacuum tubes, were used in 2nd generation computers and hence size became smaller. It was less expensive, consumed less electricity, emitted less heat, more powerful and faster. These computers used high level languages like FORTRAN (Formula translation) and COBOL (Common Business Oriented Language).

Third generation computers (1964 – 1971):

Third generation computers were smaller in size due to the use of integrated Circuits (IC’s). It reduced the size again and increased the speed and efficiency. New input/output devices like keyboard and mouse were introduced at this age.

Fourth generation computers (1971 onwards):

Fourth generation computers use Microprocessors and hence they are called as

microcomputers. Microprocessor is a single chip which contains Large Scale of Integration (LSI) of electronic components like transistors, capacitors, resistors, etc. Due to this, it became possible to place a CPU on a single chip. Later LSI were replaced by Very Large-Scale Integrated Circuits (VLSI).

Fifth generation computers (future):

Fifth generation computers are based on Artificial Intelligence (AI). AI is the ability to simulate human intelligence. AI is the branch of computer science concerned with developing intelligent systems for solving complex problems. The most common AI programming language are LISP and Prolog.

Chapter 2 – Data Representation and Boolean Algebra

Number System

A Number system defines a set of values to represent quantity.

Radix or base of a number system: The number of symbols or digits used in a number system is called radix or base of a number system. For example, Decimal number system uses 10 digits (0 to 9) therefore the base of Decimal number system is 10.

MSD (Most Significant Digit) and LSD (Least Significant Digit): The left most digit in a decimal number system is called MSD and the right most digit is called LSD.

Decimal Number System: The decimal number system involves ten symbols 0,1,2,3,4,5,6,7,8 and 9 to form a number. Its base is 10.

Binary number system: The binary number system uses the digits 0 and 1. The base of binary number system is 2. Each digit in a binary number is called bit.

Octal number system: The Octal number system uses the digits from 0 to 7. The base of Octal number system is 8.

Hexadecimal number system: The Hexadecimal number system uses the digits from 0 to 9 and the letters from A to F. The base of Hexadecimal number system is 16.

Decimal, Binary, Octal, Hexidecimal Values			
Decimal	Binary	Octal	Hexidecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Number Conversions:

To convert a decimal number into some other number systems, divide the decimal with the base of the new number system repeatedly and the remainders are noted. To get the final answer, write the remainders in reverse order.

To convert Binary / Octal / Hexadecimal numbers into Decimal number, multiply each digit / bit with its positional value and add to get the decimal equivalent.

Decimal to Binary:

Find binary equivalent of $(25)_{10}$.

2	25	Remainders
2	12	1
2	6	0
2	3	0
2	1	1
0		1

So, $(25)_{10} = (11001)_2$

Decimal to Octal:

Find Octal equivalent of $(125)_{10}$.

8	125	Remainders
8	15	5
8	1	7
	0	1

So, $(125)_{10} = (175)_8$.

Decimal to Hexadecimal:

Find Hexadecimal equivalent of $(380)_{10}$.

16	380	Remainders
16	23	12 (C)
16	1	7
	0	1

So, $(380)_{10} = (17C)_{16}$.

Binary to Decimal:

Convert $(10011)_2$ to decimal.

$$\begin{aligned}(10011)_2 &= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 16 + 2 + 1 \\ &= 19\end{aligned}$$

So, $(10011)_2 = (19)_{10}$

Octal to Decimal:

Convert $(245)_8$ to decimal.

$$\begin{aligned}(245)_8 &= 2 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 \\ &= 128 + 32 + 5\end{aligned}$$

$$= 165$$

$$\text{So, } (245)_8 = (165)_{10}$$

Hexadecimal to decimal:

Convert $(A5C)_{16}$ to decimal

$$(A5C)_{16} = 10 \times 16^2 + 5 \times 16^1 + 12 \times 16^0$$

$$= 2560 + 80 + 12$$

$$= 2652$$

$$\text{So, } (A5C)_{16} = (2652)_{10}$$

Data Representations

Data representation is the method used internally to represent data in a computer. Different types of data are: integers, floating point numbers, characters, images, audio, video etc. They can be represented as:

Representation of numbers:

Numbers can be classified into integers and floating-point numbers.

Representation of Integers

The three methods for representing integers in computer memory are:

- 1) Sign and magnitude representation
- 2) 1's Complement representation
- 3) 2's Complement representation

1. Sign and magnitude representation:

In sign and magnitude representation, leftmost bit (sign bit) in the number represents sign of the number. The sign bit is 0 for positive numbers and 1 for negative numbers.

Example: Representing +9 and -9 in sign and magnitude method

$$9 = (1001)_2$$

$$9 \text{ in 8-bit form } (0000\ 1001)_2$$

$$+9 = (0\ 0\ 0\ 0\ 1\ 0\ 0\ 1)_2$$

Sign bit. Since it is 0, the number is positive.

$$-9 = (1\ 0\ 0\ 0\ 0\ 0\ 0\ 1)$$

Sign bit. Since it is 1, the number is negative.

2. 1's Complement representation:

In this method Positive numbers are represented in their original binary Form, and negative numbers in One's complement form. 1's Complement of a negative number is obtained by the following steps, 1) Find binary form of number (without sign).

2) Provide sufficient zeroes to convert to satisfy word size (8-bit ,16 bit etc.).

3) Replace all 1's with 0's and 0's with 1's, in the binary form obtained in the step two.

Example: Represent -17 in 1's complement form

$$(17)_{10} = (10001)_2$$

Converting to 8-bit, $(17)_{10} = (00010001)_2$

$$1's\ complement\ of\ (00010001)_2 = (11101110)_2$$

(Here, every 1 is replaced by 0 and 0 by 1 to convert to 1's complement).

[Note: No complement should be taken for positive numbers. Only negative numbers are represented in complement form.]

3. 2's Complement representation:

In 2's Complement representation positive numbers are represented in binary form and negative numbers in 2's complement form. The 2's complement is obtained by adding 1 to its 1's complement.

ie, 2's Complement = 1 + 1's Complement.

Example: Represent -17 in one's complement form

$$(17)_{10} = (10001)_2$$

Converting to 8-bit, $(17)_{10} = (0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)_2$

$$\text{One's complement of } (00010001)_2 = (1\ 1\ 1\ 0\ 1\ 1\ 1\ 0)_2$$

Hence 2's Complement = 1 1 1 0 1 1 1 0 +

1

1 1 1 0 1 1 1 1

Hence, 2's complement of -17 is, (1 1 1 0 1 1 1 1)₂

Representation of Characters

Alphanumeric codes are used to represent alphabets and numbers in a computer. Some of the commonly used alphanumeric codes are:

ASCII (American Standard Code for Information Interchange): ASCII is a popular character encoding system. This code is used in data Communication. It is a 7-bit code. ASCII is of two types ASCII -7 and ASCII -8. Using ASCII - 7, 128 unique combinations or representations and by ASCII - 8, 256 combinations are possible.

Unicode: Unicode is a universal character encoding standard for representing text in any written language. It uses 16 bits. Nowadays Unicode uses more than 16 bits. Unicode has the following advantages: 1) It can represent any character in any language. 2) It is compatible with ASCII 8. 3) Unicode uses 16 bits and supports more characters than ASCII.

Introduction to Boolean Algebra

Boolean algebra is the branch of algebra in which the values of the variables are the truth values 'True' and 'False' and numerically 1 and 0, respectively. Boolean algebra is the algebra of logic, which is a part of mathematical algebra. The name Boolean is given to honour British mathematician George Boole.

Boolean operators and logic gates

For every Boolean operation, the operands are always Boolean variables or constants and the result will always be true (1) or false (0).

Computers perform logical operations with electronic circuits, called logic circuits. A Logic circuit is constructed with logic gates. A logic gate is a physical device that can perform

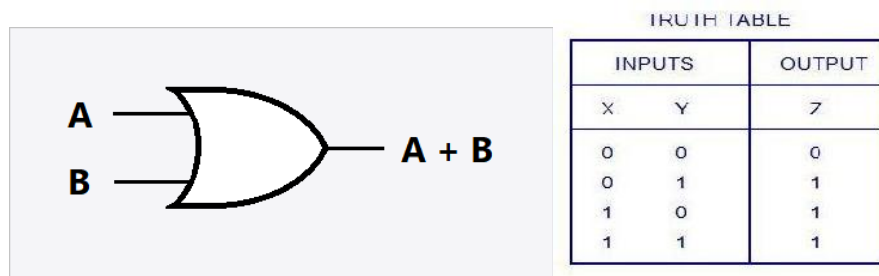
Logical operations on one or more logical inputs and produce a single output. Logic gates are implemented using diodes or transistors. Three logic gates are OR, AND and NOT.

OR – Logical Addition. AND – Logical Multiplication NOT – Logical Negation.

A truth table is a table that shows all possible inputs and their corresponding outputs for the given operation.

OR gate

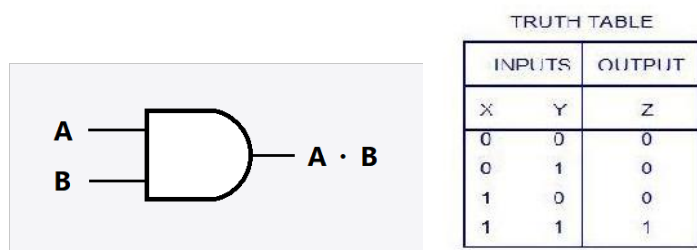
OR operator is used for logical addition. The symbol used for OR operation is +. The expression $A+B$ is read as A OR B. The logic gate used to implement the logical OR operation is called logical OR gate. Output of this gate will be 1 if any input is 1. The logical OR gate is depicted below:



AND gate

AND operator is used for logical multiplication. The symbol used for AND operation is. (dot). Output of this gate will be 1 if and only if all inputs are 1.

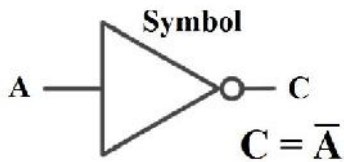
The logical AND gate is depicted below:



NOT gate

NOT operator performs logical negation. It is a unary operator since it requires only one operand. The symbol used for this operation is \bar{A} (A bar) or A' (A dash). NOT gate is also known as inverter. If input is 0, output will be 1.

NOT Gate

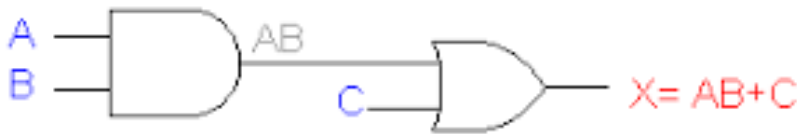


Truth Table

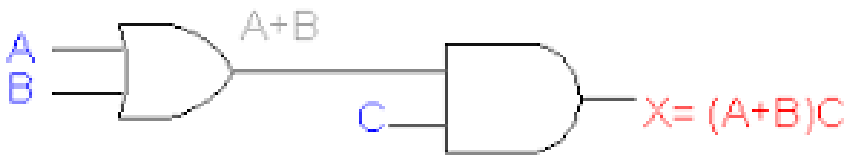
INPUT	OUTPUT
A	NOT A
0	1
1	0

Circuit designing for simple Boolean expressions:

Construct a logical circuit for Boolean expression $A \cdot B + C$



Construct a logical circuit for Boolean expression $(A + B) \cdot C$



Chapter – 3 Components of the Computer System

Hardware

The physical or tangible components of a computer is called hardware. Example: Mouse, keyboard, Monitor etc.

Processor: The processor, also known as CPU, is responsible for computing and decision making. It is sometimes called logic chip. It contains thousands of transistors. Some of the available processors are Intel core i3, i5, i7, AMD etc.

Peripherals

Peripherals are devices attached to a computer to enhance its capabilities. They communicate with motherboard through ports.

Motherboard: The Motherboard is a Printed Circuit Board (PCB) on to which components are integrated. It is the main circuit board in a computer. The physical dimension of motherboard is called form factor. External devices are connected to the motherboard through ports. Some of the ports used on computers are,

- 1) **Serial port:** A Serial port transmits one bit at a time. It is also called COM port. It is used to connect mouse, Keyboard, modem. It is much slower and is replaced by PS/2, USB.
- 2) **Parallel port:** A Parallel port transmits several bits at a time. It is faster than a serial port and is used to connect a printer or scanner.
- 3) **USB port (Universal Serial Bus):** USB provides high speed data communication. It allows multiple connections at the same port.
- 4) **LAN port:** LAN port also known as ethernet port, is used to connect a computer to a network.
- 5) **PS/2 port:** It is used to connect a mouse or keyboard to a computer. It is much slower and is replaced by USB port.
- 6) **Audio ports:** Audio port is used to connect audio devices such as speaker, microphone etc.
- 7) **VGA port (Video Graphic Array):** The VGA port is used for connecting a computer to a monitor. It was introduced by IBM.
- 8) **HDMI (High-Definition Multimedia Interface):** It is a standard for transferring High-definition video and audio. It is a replacement for existing analogue video standards.

Memory

Memory is used to store data and instructions. Memory can be classified into two: Primary memory and Secondary memory.

Primary memory holds data and results temporarily. Secondary memory on the other hand holds data and information permanently. Primary memory is volatile as secondary memory is non-volatile.

Primary memory: There are three types of Primary memory: Random Access Memory (RAM), Read Only Memory (ROM) and Cache memory.

RAM (Random Access Memory): RAM is volatile and is used to temporarily store data and instructions. Primary memory is a semiconductor memory. RAM provides a limited storage capacity such as 2GB, 4GB etc. The speed of RAM is measured in Mega Hertz (Mz).

ROM (Read Only Memory): ROM is non-volatile and is used to store start-up or bootstrap programs (Firmware).

RAM	ROM
It is faster than ROM.	It is a slower memory.
It allows reading and writing.	Allows reading only.
Volatile.	Non-volatile.
Stores data when computer is functioning.	Stores boot programs.

e-Waste: e - Waste refers to un-wanted or non-working electronic products. They cause serious health and pollution problems. They contain harmful metals such as lead, cadmium, mercury etc. Rapid changes in technology and falling price are cause for e-waste.

a) Reuse: It refers to second hand use or use after slight modification.

b) Incineration: It is a controlled and complete combustion process, in which waste materials are burned in specially designed incinerators at high temperatures. It can cause release of metals such as cadmium, lead to the atmosphere.

c) Recycling of e-waste: It is the process of manufacturing new products from e-waste. Monitors, Hard disks, laptops etc. can be recycled.

d) Land filling: It is the widely used method for the disposal e-waste. Soil is excavated and e-waste is buried in it.

Role of Students in e-waste disposal: Buy only necessary electronic goods. Repair and use if possible. Buy goods with take back policy. Avoid disposal of battery-operated devices. Recycle or donate electronic goods. Dispose electronic goods in proper way.

Software

System software: A system software is a software designed to control the hardware. It provides an environment for the working of application programs. It is classified into three: Operating system, Language Processor and Utility software.

Operating system: An Operating system is a program that acts as an interface between the hardware and the user. The primary aim of operating system is to use hardware in an efficient way. An Operating system is a software that manages the hardware and resources. Some of the commonly used Operating systems are Windows, DOS, LINUX, UNIX. Important functions of Operating systems are:

- 1) **Process management:** A process is a program in execution. An operating system allocates, deallocates and schedules processes.
- 2) **Memory management:** Allocation of main memory and storage is done by operating system. It also manages free space.
- 3) **File management:** A file is a collection of related information. The operating system handles the following file activities, · Creation and deletion of files. · Creation and deletion of directories. · Backup of files.
- 4) **Device management:** It manages devices connected to a computer.

Language processors

A language processor is a program which converts source code into machine code.

Computer languages are classified into two types: High level Languages and low-level Languages. Machine language and Assembly language are the different low-level languages.

Machine Language: The language which uses binary digits (0 and 1) is called machine language. It is also called as binary language, since it uses only 0 and 1. Here, 1 indicates ON or presence of current and 0 indicates OFF or absence of current. Since current is the language of machines, this is called machine language.

Assembly Language: An assembly language is a low-level language which uses mnemonics. The major disadvantage of assembly languages is: Programming is difficult and time consuming, It is machine dependent, Programmer should have the knowledge of hardware, Not portable.

Advantages: Faster than high level language, Efficient memory use.

High Level Languages (HLL): High level language are languages which use English like words and sentences. It is not understandable to a computer. Some examples of high-level languages are BASIC, C, C++, Java, etc.

1. **Assembler:** An assembler is a software which converts assembly language into machine language. Assembler is highly machine dependent.
2. **Compiler:** A compiler is a software which converts high level language (source code) into machine language (object code). It scans the entire program in a single run. If there is any error in the program, the compiler provides a list of errors at the end of compilation. If there are no errors, the compiler will generate object file.
3. **Interpreter:** An interpreter is a software which converts high level language into machine language. It translates and executes one statement at a time. For example, BASIC is an interpreted language. Characteristics of interpreter.
 - It translates line by line into machine code.
 - It detects only one error at a time.

Freeware and Shareware: A freeware is a software available for use free of charge for an unlimited period. A shareware is a software distributed on a trial basis. They are also called trialware.

Free and Open-source software: Free and Open-source software (FOSS) is a software that can be used freely, copy, modify and redistribute. The four essential freedoms:

1. Freedom 0: - The freedom to run the program as you wish, for any purpose.
2. Freedom 1: - The freedom to study how the program works.
3. Freedom 2: - The freedom to distribute copies.
4. Freedom 3: - The freedom to improve programs and release the improvements to the public.

Examples of free and open-source software: GIMP, Mozilla Firefox, Open Office, Linux.

Human ware or Liveware: It refers to the people who use the computer. For example: Programmers, System Analyst, Data entry operators etc. They are also called Skin ware or peopleware.

Firmware: The software permanently stored in hardware is called firmware.

Chapter 4 – Principles of Programming and Problem Solving

Phases in programming

The different stages in programming are,

1. Problem identification.
2. Preparing algorithms and flowcharts.
3. Coding.
4. Translation.
5. Debugging.
6. Execution and testing.
7. Documentation.

1. Problem solving

In this phase the problem is analysed. The data involved in processing, the output to be obtained are identified. The data involved in processing, its type and quantity, formula to be used, activities involved and the output to be obtained are identified in this stage.

2. Algorithms and Flowcharts

Once the problem is identified, a step-by-step procedure is developed to solve the problem.

Algorithm

An algorithm is a finite sequence of instructions to solve a problem. It is a step-by-step procedure to solve a problem. Famous Arab Mathematician **Abu Jafar Mohammed Ibn Musaa Al – Khowarizmi** invented first algorithm.






Characteristics of an algorithm

1. It should begin with instruction(s) to accept inputs.
2. Each instruction must be precise and should have only one meaning.
3. The number of instructions must be finite.
4. An algorithm must produce desired output.

Flowchart

A flowchart is a pictorial representation of an algorithm. It is mainly used to understand an algorithm.

Symbols used in a flowchart:

Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	Show flow of data
	Input/Output	A parallelogram represents input or output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

Advantages of flowchart

Following are the advantages of flowchart

- 1) Better communication: Flowchart are the better way of communicating the logic of a program.
- 2) Effective analysis: Effective analysis of a program can be done using a flowchart.
- 3) Effective synthesis: Flowchart are used as a working model in designing a new program.
- 4) Efficient coding: Flowchart guide the programmer in writing the code.

Limitations of flowchart

A flowchart has the following limitations,

- 1) Any change or modification in the logic requires a new flowchart.
- 2) There are no standards for drawing a flowchart.
- 3) The drawing of flowchart is a time-consuming process.

3. Coding

The set of instructions expressed in any programming language is called a computer program.

The process of writing a program is called coding.

4. Translation.



Compiler or Interpreter is used for translation. During this step, the syntax errors of the program will be displayed. The errors are to be corrected and again translation must take place.

5. Debugging

The process of detecting and correcting errors is called debugging. There are two types of error, Syntax error and Logical error.

Syntax error: Syntax error occurs due to incorrect use of program statement. Logical error is due to error in planning the logic of a program. A compiler can detect syntax error. Logical error can only be detected by a programmer. For example: undefined variable, incorrect words are example of a syntax error.

Logical error: Logical error is an error which occurred due to error in planning the logic of a program. Compiler cannot identify this.

Run time error: Errors that occur during execution of a program is called run time error. For example, 'Division by Zero' is a run time error.

Note: A program will execute even if there is a logic error, but the output will be wrong.

Execution and testing

The purpose of testing is to find if the results are correct. The program will be executed and the result will be compared. The accuracy of the program logic can be determined by testing.

Documentation

This is the last step in programming. Writing comments in source code is called internal documentation. It helps in program modification later.

Another form of documentation is preparing user manual. They are hard copy documents and are called external documentation.

Chapter 5: Introduction to C++ Programming

C++ is a powerful, popular and object-oriented programming language developed by Bjarne Stroustrup.

Character set: Character set is the set of valid symbols that C++ compiler can recognize. A character represents any letter, digit, alphabet or any other symbol. The character set of C++ is categorized as

- i) Letters : (A – Z, a – z)
- ii) Digits : (0 – 9)
- iii) Special characters : +, -, *, /, /, \,

iv) White spaces : Space bar, Tab, Carriage return, New line, Form feed etc.

v) Other characters: Any other characters, other than Special characters.

Tokens:

Tokens are the fundamental building blocks of the program. They are also called as lexical units. C++ has five types of tokens – Keywords, Identifiers, Literals, Punctuators and Operators.

1. Keywords: keywords are the words (tokens) that convey a specific meaning to the language compiler. These are also known as **reserved words** or special words as they are reserved for special purposes. Eg:- int, for, if, main....

2. Identifiers: These are the user-defined words used to name different program elements such as memory locations, statements, functions, objects, classes etc. The identifiers of memory locations are called variables. The identifiers assigned to statements are called labels. The identifiers used to refer a set of statements are called function names.

Rules for naming identifiers:

- An arbitrary long sequence of letters, digits and underscores.
- The first character must be a letter or underscore (_).
- White space and special characters are not allowed.
- Keywords cannot be used.
- Upper- and lower-case letters are treated differently.

3. Literals: Data items that never change their value during the program run. They are often referred to as constants. Literals are of four types – Integer literals, Floating point literals, Character literals and String literals.

Integer literals: The tokens constituted only by digits. They are whole numbers without fractional part. Eg: 13, -76

Floating point literals: Also known as real constants. These are numbers having fractional parts. These can be written in one of the two forms –fractional form or exponential form.

Eg: 4.35, 2.564, -9.8, -0.5 etc.

Character literals: A single character enclosed in single quotes. Eg: ‘a’, ‘9’, ‘+’ etc.

Escape sequences: These are special character constants that represent non-graphic symbols. It consists of a backslash (\) followed by one or more specific characters.

Some common escape sequences and their non-graphic characters are:

\a : audible bell. \b : ack space. \n : new line. \t : horizontal tab.

\0 : Null character.

String constant: A sequence of one or more characters enclosed within a pair of double quotes is called. Eg: “Hello friends”, “123” etc.

4. Punctuators: Some special symbols that have syntactic or semantic meaning to the compiler.

5. Operators: A symbol that tells the compiler about a specific operation. They are the tokens that trigger some kind of operation. The operator is applied on a set of data called operands.

Chapter 6 – Data Types and Operators

Data types: These are the means to identify the nature of the data and the set of operations that can be performed on the data. Various data types are defined in C++ to differentiate data characteristics.

C++ provides a rich set of data types. Based on nature, size and associated operations, data types are classified as: **Fundamental datatypes**, **Derived datatypes** and **User defined datatypes**.

1. Fundamental datatypes: These are the built-in datatypes of C++. These are also known as basic datatypes or atomic datatypes. They are atomic in nature and cannot be further decomposed of. The five fundamental datatypes in C++ are char, int, float, double and void.

2. Derived datatypes: Derived data types are constructed from fundamental datatypes. Derived datatypes are: Array, pointers, functions etc.

3. User Defined datatypes: User defined datatypes are datatypes which are defined by the user himself. User can define the following datatypes: Structure, Union, enumeration, class etc.

Fundamental datatypes: These are the built-in datatypes of C++. The five fundamental datatypes in C++ are char, int, float, double and void.

1. int datatype: Integers are whole numbers without a fractional part. They can be positive, negative or zero. C++ allows 4 bytes of memory for integers.

2. char datatype: Characters are the symbols covered by the character set of C++. The characters are internally treated as integers. Only 1 byte of memory is required for char datatype.

3. float datatype: Numbers with a fractional part are called as float datatype. C++ allows 4 bytes of memory for float data types.

4. double datatype: double datatypes are also numbers with fractional parts. They have bigger size compared to float. Its size is 8 bytes.

5. void datatype: The datatype void is a keyword and it indicates an empty set of data. It does not require any memory space.

Variables: The names given to memory locations, by which data in these locations are referenced. The value stored in the location is called content of the variable.

Operators: The tokens or symbols that trigger computer to carry out operations. The data on which an operation is taken place is called operands. An operand may be either a constant or a variable. Operators are classified into three: **unary** (operates on 1 operand), **binary** (operates on 2 operands) and **ternary** (operates on 3 operands). Operators in C++ are:

1. Arithmetic operators: These are used to perform basic arithmetic operations such as:

Addition (+), subtraction (-), multiplication (*), division (/), modulus (%)

2. Relational operators: Relational operators are used for comparing numeric data. These are binary operators. The result of any relational operation will be either True or False.

Less than (<), Greater than (>), Less than equal to (< =), Greater than equal to (> =),

Equal to (= =), Not equal to (! =)

3. Logical Operators: Logical operators are used to combine two or more relational expressions. C++ provides three logical operators and they are:

Logical AND (&&) - The result will be true if both the expressions are true; false otherwise.

Logical OR (||) - The result will be false if both the expressions are false; true otherwise

Logical NOT (!) - Negates the logical value of an expression.

4. Assignment Operator:

Assignment operator (=) is used to assign or store a value in a memory location.

5. Arithmetic assignment operators:

We can write $a = a + 10$ as $a += 10$ in a short hand method. The arithmetic assignment operators in C++ can be used as:

$x += 10 ;$ \longrightarrow $x = x + 10 ;$

$x *= 10 ;$ \longrightarrow $x = x * 10 ;$

6. Input/output operator:

C++ provides $>>$ operator for input operation. The operator is known as **get from or extraction operator**. The operator $<<$ is used for output operation and is called **put to or insertion operator**.

Expressions:

Expressions are composed of operators and operands. Expressions are classified into arithmetic expressions, relational expressions and logical expressions.

Arithmetic expression: This is an expression in which only arithmetic operators are used. They are further classified into integer expressions and floating point (real) expressions.

Integer expression: An arithmetic expression that contains only integer operands and produces an integer result after performing all the operations is an integer expression.

Floating point or real expression: An arithmetic expression that contains only floating-point data and returns a floating-point result after performing all the operations is a floating-point expression.

Relational expression: This is an expression that contains relational operators. It produces Boolean type results like True (1) or False (0).

Logical expressions: this is an expression that uses logical operators to combine two or more relational expressions. It produces either True or False as the result.

Type conversion: Type conversion is conversion of the data type of an operand into another type. It is done in two ways: implicitly and explicitly.

Implicit type conversion or Type promotion: It is the implicit type conversion, which is performed by C++ compiler internally. C++ converts the lower sized operands to the data type of highest sized operand. The conversion is always from lower type to higher and hence the name type promotion.

Explicit type conversion or Type casting: It is the explicit type conversion and is done by the programmer by specifying the data type within parentheses to the left of the operand.

Eg:- `sum = (float) 5 + 7 ;` results 12.0

Statements

Statements are the smallest executable unit of a programming language. C++ uses; (semicolon) as the delimiter of a statement. Different types of C++ statements are:

1. Declaration statements: Any variable must be declared before it is used. When we declare a variable, we tell the compiler about the type of data that will be stored in it.

Variable declaration syntax: **datatype <variable1>, <variable2>, <variable3>, . . . ;**

2. Assignment statements: If in a statement, an assignment operator is used, it is called assignment statement.

3. Input statements: Input statement is a means that allows the user to store data in the memory during the execution of the program. The get from or extraction operator (>>) specifies the input operation. Eg: `cin>>num;`

4. Output statements: output statements make the results available to users through any output device. The put to or insertion operator (<<) is used to specify this operation.

Chapter 6 – Control Statements

Decision making statements

The statements provided by C++ for the selected execution are called decision making statements or selection statements. if and switch are the two types of selection statements in C++.

if statement

The if statement is used to select a set of statements for execution based on a condition. In C++, conditions, otherwise known as test expressions are provided by relational or logical expressions. The syntax (general form) of if statement is as follows:

```
if (test expression)
{
    statement block;
}
```

Here, the test expression represents a condition which is either a relational expression or logical expression. If the test expression evaluates to True (non-zero value), a statement or a block of statements associated with if is executed. Otherwise, the control moves to the statement following the if construct.

if...else statement

If we want to execute some actions when the condition becomes False, we introduce another form of if statement, if..else. The syntax is:

```
if (test expression)
{
    statement block 1;
}
else
{
```

```
        statement block 2;  
    }  
}
```

Here, if the test expression evaluates to True, only the statement block 1 is executed. If the test expression evaluates to False, statement block 2 is executed.

Nested if

In some situations, there may arise the need to take a decision within if block. When we write an if statement inside another if block, it is called nesting. Nested means one inside another.

The important point to remember about nested if is that an else statement always refers to the nearest if statement within the same block.

The else if ladder

There are situations where an if statement is used within an else block. It is used in programs when multiple branching is required. Different conditions will be given and each condition will decide which statement is to be selected for execution. A common programming construct based on if statement is the else if ladder, also referred to as the else if staircase because of its appearance. It is also known as if...else if statement.

switch statement

switch statement successively tests the value of a variable or an expression against a list of integer or character constants.

The syntax of switch statement is given below:

```
switch (expression)  
{  
    case constant_1 : statement block 1; break;  
    case constant_2 : statement block 2; break;  
    :  
    default : statement block n;  
}
```

In the syntax switch, case, break and default are keywords. The expression is evaluated to get an integer or character constant and it is matched against the constants specified in the case statements. When a match is found, the statement block associated with that case is executed until the break statement or the end of switch statement is reached. If no match is found, the statements in the default block get executed. The break statement, used inside switch, is one of the jump statements in C++. When a break statement is encountered, the program control goes to the statements following the switch statement.

switch statement	if else if ladder
Permits multiple branching	Permits multiple branching.
Evaluates conditions with equality operator only.	Evaluate any relational or logical expression.
Case constant must be an integer or a character type value.	Condition may include range of values and floating-point constants.

The conditional operator (? :)

The conditional operator (?:) is a ternary operator consisting of symbols ? and : (a question mark and a colon). It requires three operands to operate upon. It can be used as an alternative to if...else statement.

Its general form is:

Test expression ? True_case code : False_case code;

The Test expression can be any relational or logical expression and True_case code and False_case code can be constants, variables, expressions or statement. The operation performed by this operator is shown below with the help of an if statement.

```

if (Test expression)
{
    True_case code;
}
else
{
    False_case code;
}

```

}

The conditional operator works in the same way as if...else works. It evaluates the Test expression and if it is True, the True_case code is executed. Otherwise, False_case code is executed.

Iteration statements

While writing programs, we use some specific constructs of the language to perform the repeated execution of a set of one or more statements. Such constructs are called iteration statements or looping statements. In C++, we have three iteration statements and all of them allow a set of instructions to be executed repeatedly when a condition is True.

four elements of a loop:

1. **Initialisation:** Before entering a loop, its control variable must be initialised. During initialisation, the loop control variable gets its first value. The initialisation statement is executed only once, at the beginning of the loop.
2. **Test expression:** It is a relational or logical expression whose value is either True or False. It decides whether the loop-body will be executed or not. If the test expression evaluates to True, the loop-body gets executed, otherwise it will not be executed.
3. **Update statement:** The update statement modifies the loop control variable by changing its value. The update statement is executed before the next iteration.
4. **Body of the loop:** The statements that need to be executed repeatedly constitute the body of the loop. It may be a simple statement or a compound statement.

Loops are generally classified into entry-controlled loops and exit-controlled loops. In entry-controlled loops, the condition is checked before the execution of the body. Whereas in exit-controlled loops, the condition is checked after the execution of the body. C++ provides three loop statements: while loop, for loop and do...while loop. In these loops, for and while loops are entry-controlled loops and do while loop is an exit-controlled loop.

while loop or while statement

The while loop is an entry-controlled loop. The condition is checked first and if it is found True the body of the loop will be executed. That is, the body will be executed as long as the condition is True.

The syntax of while loop is:

```
initialisation of loop control variable;

while (test expression)
{
    body of the loop;
    updation of loop control variable;
}
```

Here, the test expression defines the condition which controls the loop. The body of the loop may be a single statement or a compound statement or without any statement. The body is the set of statements for repeated execution. Update expression refers to a statement that changes the value of the loop control variable. In a while loop, a loop control variable should be initialised before the loop begins and it should be updated inside the body of the loop.

for loop or for statement

The for loop is also an entry-controlled loop in C++. All the three loop elements (initialisation, test expression and update statement) are placed together in for statement. So, it makes the program compact.

The syntax is:

```
for (initialisation; test expression; update statement)
{
    body-of-the-loop;
}
```

The execution of for loop is the same as that of while loop. Since the three elements come together, this statement is used more in situations where counting is involved.

do...while statement

In the case of for loop and while loop, the test expression is evaluated before executing the body of the loop. If the test expression evaluates to False for the first time itself, the body is never executed. But in some situations, it is necessary to execute the loop body at least once,

without considering the result of the test expression. In that case the do...while loop is the best choice.

Its syntax is:

```

initialisation of loop control variable;

do
    {
        body of the loop;

        Updation of loop control variable;
    } while (test expression);

```

do...while loop is an exit-controlled loop

for loop	while loop	do...while loop
Entry controlled loop	Entry controlled loop	Exit controlled loop
Initialisation along with loop definition	Initialisation before loop definition	Initialisation before loop definition
No guarantee to execute the loop body at least once	No guarantee to execute the loop body at least once	Will execute the loop body at least once even though the condition is False

Jump statements

The statements that facilitate the transfer of program control from one place to another are called jump statements. The two types of jump statements are: break and continue.

break statement: When a break statement is encountered in a program, it takes the program control outside the immediate enclosing loop or switch statement.

continue statement: continue statement is a jump statement used for skipping over a part of the code within the loop body and forcing the next iteration. The continue statement forces next iteration of the loop.

Chapter 8 – Arrays

Arrays are used to store a set of values of the same type in contiguous memory locations under a single variable name. Each element in an array can be accessed using its position in the list, called index number or subscript.

Declaring arrays: Like all other variables, array also must be declared properly before it is used.

The syntax for declaring an array is: `data_type array_name [size];`

Eg:-

- a. To store scores of 100 students: `int score [100];`
- b. To store English letters: `char a [10];`

Array Initialisation: Giving values to the array elements at the time of array declaration is known as array initialization. Array can be initialised by listing values within braces, separated by comma.

Eg: `int num [5] = {78, 89, 93, 88, 90};`

Initial values are stored in the order they are written.

Memory allocation for arrays: The amount of storage required to hold an array is directly proportional to its type and size. The memory space allocated for an array can be computed using the formula: `total_bytes = size of array_type * size-of_array.`

For eg:- Total bytes allocated for the array declared as `float num [15]` will be $4 * 15 = 60$ Bytes.

Accessing elements of arrays: Array elements can be accessed using their subscript. In C++, array index starts with 0. If an array is declared as `int num[10]`; then the possible index values are from 0 to 9. The first element can be referred as `num[0]`, second element as `num[1]` and so on.

Array Operations: The operations performed on arrays include traversal, searching, insertion, deletion, sorting and merging.

i. Traversal- Traversal means accessing each element of the array at least once. Displaying all the elements of an array.

ii. Sorting - Sorting is the process of arranging the elements of the array in some logical order. This logical order may be ascending or descending.

iii. Searching - Searching is the process of finding the location of the given element in the array.

iv. Insertion - Insertion is the process of adding new elements into the array

v. Deletion - Deletion is the process of removing existing elements from the array.

vi. Merging - Merging is the process of combining two or more arrays into a single resultant array.

Array Traversal: Accessing each element of an array at least once to perform any operation is known as traversal operation. Eg. Displaying all elements of an array.

Program for array traversal

```
#include <iostream>

using namespace std;

int main ()
{
    int a [5], i;

    cout<<"Enter the elements of the array:";

    for (i=0; i<5; i++)

    cin >> a[i];           //Reading the elements

    for (i=0; i<5; i++)

    a[i] = a[i] + 10;      // adds 10 to all elements in array, a case of traversal.

    cout<<"\n Now value of elements in the array are...\n";

    for (i=0; i<5; i++)

    cout<< a[i]<< "\t";   // Another case of traversal

    return 0;

}
```


Chapter 09 - String Handling And I/O Functions

String handling using arrays

String is a kind of literal, written as a sequence of characters within a pair of double quotes. So, string is declared as an array of type char. ie, `char string_name [size];`

A string can be initialised as: `char name [10] = {'c', 'o', 'm', 'p', 'u', 't', 'e', 'r'};`

A null character ' \0 ' is stored at the end of every string. This character is used as the string terminator and added at the end automatically. So, in every string, one Byte is reserved to store null character.

Memory allocation for strings

We know, string is an array of characters and a character needs one byte of storage space. So, the memory required to store a string will be equal to the number of characters in the string plus one byte for null character.

For eg- `char name [10];`

Word =” Computer”; can be represented in memory as:

C	o	m	p	u	t	e	r	\0	
---	---	---	---	---	---	---	---	----	--

Input output operations in strings.

`cin>>` statement is used to store a string without spaces. A white space is treated as a separator of data. So, `cin>>` is incapable to read white spaces. This problem can be solved by using `gets ()` function. `gets ()` function is used to accept a string of characters including white space from a standard input device (eg. Keyboard) and store it in a character array.

Syntax: `gets (String_data);`

Eg. `gets (str);`

Similarly, `puts ()` function is used to display a string data on a standard output device (eg. Monitor)

Syntax: `puts (String_data);`

Eg. `puts (“Hello “);`

Header file **cstdio** must be included in the program for `gets` and `puts` functions.

Stream functions for I/O operations

C++ supports a number of input/output operations. The stream acts as an intermediate between I/O devices and the user. A stream is a group of bytes. Streams are available in the form of functions stored in iostream header file.

Input functions

Input functions allow us to input character and string data. The `get ()` and `getline ()` functions allow us to accept a string or a character from input object into memory.

Following are the stream input functions in C++,

a) `get ()`: The `get ()` function accepts a single character or string through the keyboard.

Eg:- `ch=cin.get (); //accepts a character and stores in ch`

OR

`cin.get (ch); //accepts a character and stores in ch.`

b) `getline ()`: The `getline ()` function is used to accept a string through the keyboard. It reads a string until a newline character is read. The format of `getline ()` function is:

`cin.getline (line,size);`

Where, `line` is the name of the variable which stores line of text and `size` specifies the maximum number of characters which can be stored in `line`.

Example: `cin.getline (name,40);`

Output functions

The output functions allow flow of bytes(stream) from memory to output object. Following are the output functions in C++.

a) `put ()`: The `put ()` function is used to display a single character.

Example: `char ch = 'c';`

`cout.put (ch); // Displays 'c' on the screen.`

b) `write ()`: The `write ()` function is used to display a string.

Example: `cout.write (str,10); //Displays " hello " on the screen.`

Chapter 10 – Functions

Concept of modular programming

Modular programming is the act of designing and writing programs as collection of subprograms. In this method, a large program is broken down into sub-programs (Modules). The process of breaking large programs into smaller sub programs is called modularization. The sub programs are also called functions. Two methods are used for modular programming: Top -Down and Bottom-Up method.

Advantages of Modular programming

Following are the advantages of Modular programming:

- a) Reduction in program size:** Modular approach helps to isolate repeated task. Instructions are grouped together to form functions and is invoked by using its name. Thus, program size is reduced.
- b) Less chance of error:** When size of program is reduced, syntax error will be less. Modularity helps to reduce logical errors also.
- c) Reduces programming complexity:** Modularity helps to reduce program complexity by dividing larger programs into smaller functions.
- d) Improves re-usability:** Once a function is written, it can be used later. It helps to reduce development time.

Dis-advantages of Modular programming

Modular programming requires more time and memory.

Functions in C++

A function is a collection of statements that performs a specific task. Functions are classified into two: built-in or predefined functions and user defined functions. Built-in functions are readily available to the users and are stored in header files. User defined functions are written by the user for specific tasks.

Predefined functions

Built-in or pre-defined functions are readily available to the users and are stored in header files. They are a part of C++ library. The values(data) passed into a function are called arguments or parameters. The result returned (given) by a function is called return value. Some functions do not return any value, whereas some functions return value. some of the commonly used predefined functions are.

I. String functions

A string is a group of characters. String functions are used to manipulate a string. The header file `<cstring>` is used for these functions.

i) strlen (): The `strlen()` function is used to find the length of a string(number of characters in the string).

The syntax is: `int strlen(string);`

ii) strcpy (): The `strcpy ()` function is used to copy one string into another.

The syntax is: `strcpy(string1, string2); // Copies string2 to string1`

iii) strcat (): The `strcat ()` function is used to join (concatenate) one string to another string. The length of the resultant string is the total length of the two strings.

The syntax is: `strcat(string1, string2);`

iv) strcmp (): The `strcmp ()` function is used to compare two strings.

The syntax is: `strcmp (string1, string2);`

Compares string1 with string2.

The function returns: 0 if both strings are equal.

-ve value if string1 is alphabetically lower than string2.

+ve value if string1 is alphabetically higher than string2.

v) strcmpi (): The `strcmpi ()` function is used to compare two strings ignoring cases. The function will treat both the upper case and lower-case letters as same for comparison. The syntax is same as `strcmp ()` function.

The function returns: 0 if the two strings are identical(same).
 a negative value if string1<string2.
 a positive value if string1>string2.

II. Mathematical functions

The commonly used mathematical functions in C++ are,

i) abs (): It returns the absolute value of an integer.

Example: `abs (-25); // Returns 25`

ii) sqrt (): It returns the square root of a number.

- Example: `sqrt (36); // Returns 6.`

iii) pow (): It is used to find power of a number.

Example: `pow (2,3); // Returns 8 (ie, 2*2*2 = 8).`

iv) sin (): it is a trigonometric function and it finds the sine value of an angle.

Syntax: `double sine (double);`

v) cos (): it is a trigonometric function and it finds the cosine value of an angle.

Syntax: `double cosine (double);`

Note: -The header file **<cmath>** is to be used for the above mathematical functions.

III. Character functions

Character functions are used to perform various operations on characters. The header file **<cctype>** supports these functions. The commonly used character functions in C++ are:

i) isupper (): This function checks if a character is in upper case or not. The function returns 1 if the character is in upper-case, and 0 otherwise.

ii) islower (): This function checks if a character is in lower case or not. The function returns 1 if the character is in lower-case, and 0 otherwise.

iii) isalpha (): This function is used to check whether the given character is an alphabet or not. The function returns 1 if the character is an alphabet, and 0 otherwise.

iv) isdigit (): This function is used to check whether the given character is a digit or not. The function returns 1 if the character is a digit, and 0 otherwise.

v) isalnum (): This function is used to check whether a character is alphanumeric or not. The function returns 1 if the character is alphanumeric, and 0 otherwise.

vi) toupper (): This function is used to convert the given character into its upper-case.

vii) tolower (): This function is used to convert the given character into its lower-case.

User-defined functions

A user defined function is a group of statements to perform a specific task. Every function will contain 3 elements.

1. Function prototype: Function prototype models the actual function. It specifies the compiler about the return type, function name and the type and number of arguments before the use of function is made.

Its syntax is: `return_type function_name(argument_list);`

2. Function definition. A function definition consists of function header and function body. The function header specifies the type of value it returns, function name and arguments.

The general format of a function definition is:

```
return_type function_name(argument_list)
{
    statements;
}
```

After execution, the function may or may not return values. But the important thing is that, a function can return only a single value.

3. Function call. A function which is defined cannot be executed by itself. It is invoked by another function such as `main ()`. Invoking a function whenever it is needed is called function call.

Arguments: Arguments are the set of values passed to a function. Depending on type of communication, functions are of four types:

The return statement returns a value to the calling function and transfers the program control back to the calling function.

Actual and Formal parameters

The values which are passed into a function are called arguments or parameters. The arguments or parameters present in function definition or called function is called formal parameter and those present in function call or calling function is called actual parameter.

Note: The number and type of actual and formal parameters must be the same.

Methods of calling functions

There are two methods of calling functions based on the method of passing arguments:

1. Call by Value method.
2. Call by Reference method.

Call by value (Pass by value) method

This is the default argument passing method in C++. In this method, the values contained in the actual parameters are passed to the formal parameters ie, copies of the actual parameters are passed to the function. Hence, any changes made to the formal parameters will not affect actual parameters.

Call by reference (Pass by reference) method

In this method, a reference of the actual parameter is passed to the function. The actual parameter and formal parameter share the same memory location. Here, any changes made to the formal parameter affects or is reflected back to the actual parameter.

Difference between **Call by value** and **Call by reference** methods.

Call by Value Method	Call by Reference Method
<ul style="list-style-type: none"> • Ordinary variables are used as formal parameters. • Actual parameters may be constants, variables or expressions. • The changes made in the formal arguments do not reflect in actual arguments. • Exclusive memory allocation is required for the formal arguments. 	<ul style="list-style-type: none"> • Reference variables are used as formal parameters. • Actual parameters will be variables only. • The changes made in the formal arguments do reflect in actual arguments. • Memory of actual arguments is shared by formal arguments.

Chapter – 11 Computer Networks

Computer Networks

Computer Network is a group of computers and other computing hardware devices connected to each other electronically through communication medium.

Advantages of computer networks

Following are the advantages of networks.

- 1. Data communication is possible:** Computer network helps user to communicate with any other user of the network through its services like e-mail chatting etc.
- 2. Resource Sharing:** The sharing of available hardware and software resources (like programs, printers, hard disk etc.) in a computer network is called resource sharing.
- 3. Reliability:** A file can have copies in different computers. So, failure of one system will not cause data loss.
- 4. Scalability:** Computing and storage capacity can be increased or decreased easily by adding/removing computer or storage devices to the network.
- 5. Price – Performance ratio:** Sharing of hardware and software instead of purchasing saves a lot of money.

Some terms related with networks

- a) Bandwidth:** It is the amount of data transfer through a communication medium in a unit of time.
- b) Noise:** It is the unwanted electrical or electromagnetic energy that lowers the quality of the data signals.
- c) Node:** Any device that is directly connected to a computer network is called a node.

Data communication

Data communication is the exchange of digital data between any two devices through a transmission medium.

Basic elements of a data communication system are:

- a) **Message:** It is the information to be communicated like text, picture etc.
- b) **Sender (transmitter or source):** The device used for sending data
- c) **Receiver:** The device that receives the message
- d) **Medium:** It is the path through which message travels from sender to receive.
- e) **Protocol:** The rules under which transmission takes place between sender and receiver.

Data communication devices

Data communication devices act as an interface between computer and the communication channel. They are used to transmit, receive, amplify and route the signals across the communication media.

a) Network Interface Card (NIC): It enables a computer to connect a network and communicate. It can break up data into small units, translate the protocols, send and receive data. It may be wired (Ethernet) or wireless (Wi-Fi).

b) Hub: Hub is used in wired networks to connect devices of the same network. It transmits data to all the devices connected to it. Only the device to which data is assigned responds to it. This increases traffic density and it reduces bandwidth.

c) Switch: Switch can be considered as an intelligent hub. It stores addresses of all the devices connected to it. Switch reads the destination address of the data from the packet and sends data to the particular destination device only. So, network traffic can be reduced.

d) Repeater: They are used to receive the incoming signal, amplify it to their original strength and retransmit it.

e) Bridge: It is used to interconnect different segments of an existing network. Only those packets addressed to a node on a particular segment allowed to pass the bridge and transmitted to all the nodes in the segment.

f) Router: It can interconnect two networks of the same type using the same protocol. It can find the best path for data packets to travel and can reduce the amount of traffic in the network.

g) Gateway: It can interconnect two different networks using the different protocols. It can translate one protocol to other and can understand the address structure used in different networks.

Data Terminal Equipments:

Data Terminal Equipments: These are the devices that control data flowing to or from a computer. These are connected at the end of the communication link.

a) Modem: It is used for communication between computers through telephone lines. It converts digital signals received from a computer into analogue signals (Modulation) and analogue signals received from telephone lines into digital signals (Demodulation). Its speed (which is denoted as bits per second-bps) determines the speed of the data communication.

b) Multiplexer/Demultiplexer: It divides the transmission medium into several logical frequency channels through which we can send many different signals at a time. At the destination, a demultiplexer separates different signals.

Topology

Topology is the way in which computers are physically interconnected to form a network.

Common types of topologies are:

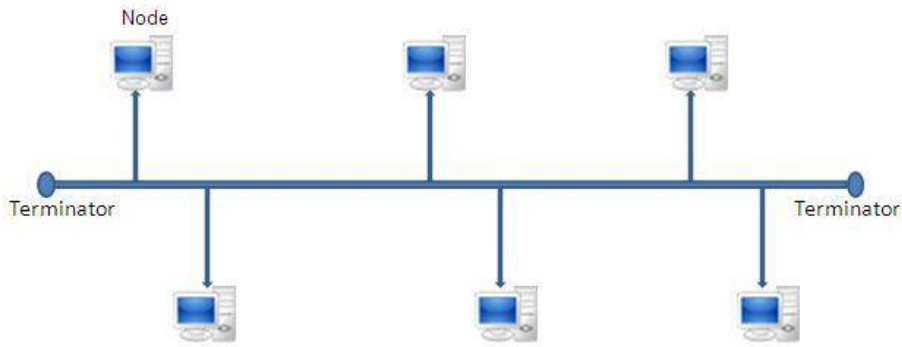
a) Bus Topology: There is a main cable called bus from the server to which every node is connected by short drop cables. A small device called terminator is attached at the end of the bus. When a data signal reaches the terminator at the end, it is absorbed and the bus is free to carry new signal.

Advantages:

- i) Easy to install.
- ii) Less cable is needed. So less expensive.
- iii) Failure of node does not affect the network.

Disadvantages:

- i) Fault detection is difficult.
- ii) Failure of cable, server or terminator will affect entire network.



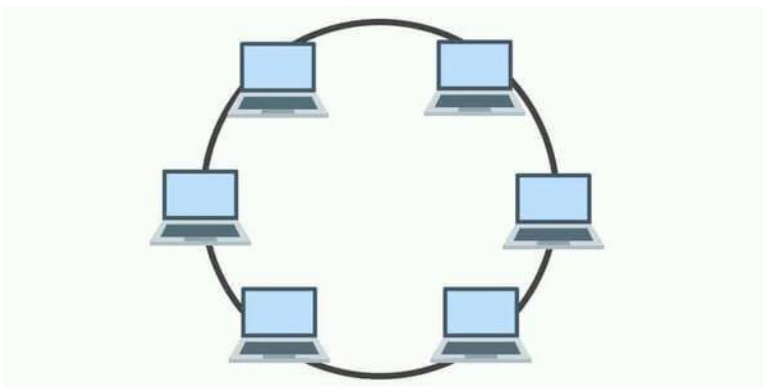
b) Ring Topology: All node computers are connected to a circular cable. All data are passing through this cable.

Advantages:

- i) No signal amplification required because each node will do it.
- ii) Requires less cable, so cost effective.

Disadvantages:

- i) If a node fails, entire network will fail.
- ii) Addition of nodes is difficult.



c) Star Topology: In star topology, there is hub/server at its centre and all other work stations are connected to it through separate connections. All messages are passed through the server or hub. When a message goes from one computer to another, it is first sent to the server, which then retransmits the message to the destination computer.

Advantages:

- i) If one workstation fails, it does not affect the whole network.
- ii) Easy to install.

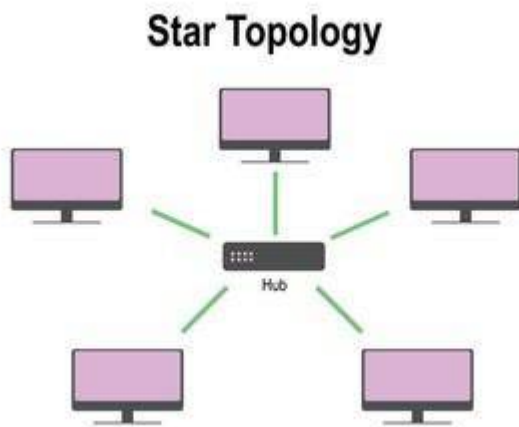
iii) It is easy to expand.

iv) Easy to find faults and remove workstations.

Disadvantages:

i) Requires more cables than Bus topology.

ii) If the central device fails it affects entire network.



d) Mesh Topology: In this Topology each node is connected to other nodes. So, there is more than one path between two nodes. Here, failure of one node may not affect the data communication.

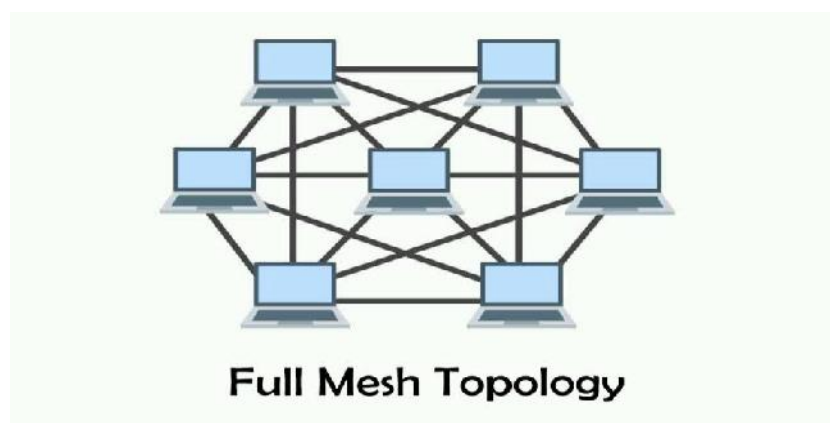
Advantages:

i) If one workstation or a path fails, it does not affect the whole network.

Disadvantages:

i) Requires more cables, so very expensive.

ii) Very complex and difficult to manage.



Identification of computers over a network.

In a network, different computers are identified by MAC address and IP address.

MAC address

Media Access Control Address is universal, unique and permanent address (12-digit hexadecimal number) assigned to each NIC by its manufacturer. Its first half contains the ID of the manufacturer and second half is the serial number of the particular adapter. The usual format of MAC address is:

MM: MM: MM: SS: SS: SS or **MM – MM – MM – SS – SS – SS**

The first half of a MAC ID (MM:MM:MM) contains the ID number of the manufacturer and second half (SS: SS: SS) represents the serial number of the device.

IP address

It is a unique 4-part numeric address assigned to each node on a network for unique identification of them. It is normally expressed in “dotted decimal Number”.

Eg. 192.165.1.1

There are two types of IP addresses:

IPV4: A 32-bit address which can identify only 4 billion devices in the net.

IPV6: A 128-bit address which can identify 4x4x4 billion devices in the net.

Uniform Resource Locator (URL)

Every network resource in the internet has a unique address known as Uniform Resource Locator (URL). It is the Address of the website or server. It consists of letters, numbers and punctuations. Eg. <http://www/dhsekerala.gov.in/model question.html>

A URL is divided in to 3 parts. They are:

a) Network protocol(scheme) like ‘http’: It enables the browser to know what protocol is used to access the information specified in the domain.

b) Domain Name: (Host name/ address) like ‘dhsekerala.gov.in’. It is the name assigned to a server through the DNS to identify a particular web server. It may contain country specific domain names like .in, .us, .au etc. and generic domain names like .com,.edu etc.

c) **File name:** It is the file to be opened like ‘model question.html’.

Chapter - 12 Internet and Mobile Computing

Services on the internet: Some of the services provided by internet are:

1) World Wide Web (WWW): The www is a collection of web servers, which contain several web pages. These web pages contain hypertext, images, video and graphics. Each web page on the internet has a unique address called URL (Uniform Resource Locator). A browser is used to access these web pages.

2) Web browser: Web browser is a software that we use to retrieve or present information and to navigate through web pages in the World Wide Web. The document to be displayed is identified using a URL. A URL consists of its DNS name and the file name to be retrieved. Web browsing refers to visiting web pages using a web browser. It is also known as surfing.

3) Search engine: A search engine is a software program used to search information on the World Wide Web. The information may be text, images etc. Some of the popular search engines are Google, Yahoo, Bing, Ask etc. Working of Search engines use a program called crawlers or spiders or robots to search the web.

4) E – mail: E- mail stands for electronic mail. It is a method of exchanging digital message over the internet. It is a popular method of communication, which enables to send images, text, video etc. as attachment along with e-mail. The e-mail send to a person is stored in his inbox folder. An e-mail address has the following structure username domain name.

Different Protocols used in e-mail: SMTP (Simple Mail Transfer Protocol): To send e-mail. MIME (Multipurpose Internet Mail Extension): To send non-text e-mail. POP3(Post Office Protocol) and IMAP (Internet Message Access Protocol): To download e- mail from mail server.

Advantages of e-mail: The following are the advantages of using e-mail,

1. Speed: An email is delivered instantly.
2. Easy to use: It is easy to send and receive e-mail.
3. Cost-Effective: It is cheaper than other forms communication.

4. Easy to reference: Sent and received e-mail messages can be stored safely.
5. Environment friendly: E-mail avoids use of paper hence saves trees.
6. Available anywhere at any time.

Dis-advantages of e-mail: Viruses are easily spread via e-mail attachments. To read –mail, one need to access internet. In-box may be filled with junk mails.

Structure of an e-mail: An e-mail has the following sections,

To: The e-mail address of recipient.

Cc (Carbon copy): The e-mail address of secondary recipient.

Bcc (Blind carbon copy): The e-mail address to which blind carbon copies are to be sent.

From: The e-mail address of the creator of message.

Reply To: The e-mail address to which replies are to be sent.

Subject: The subject of message.

Content: It contains actual message.

5) Social media: social media refers to the use of mobile and web-based technologies through which individuals and communities can create, share, discuss and modify content. In social media, interactions among people happen in virtual communities and networks over Internet. These digital technologies influence the formation and activities of civil communities to a great extent.

Classification of social media: The different types of social media are Internet forums, social blogs, microblogs, wikis, social networks, content communities etc.

- 1. Internet forums:** It is an online discussion site where people can take part in conversation in the form of posted messages. It is also known as message board. Example is Ubuntu forum.
- 2. Social blogs:** A blog is an informational site published on the internet. A blog combines text, images and other links together. Blogger is the most popular blogging services today. Wordpress.com and Blogger.com are popular sites that offer social blogs.

3. Microblogs: Microblogging is the practice of posting small pieces of digital information. The contents of a microblog are small often called micro posts. Twitter.com is a popular micro blogging site.

4. Wikis: A Wiki is a web application that allows the peoples to add, modify or delete contents. Wikipedia is the most popular wiki. It differs from a blog in that the contents of wiki are created by many users.

5. Social networks: A social network allows people to build web pages and connect and share the contents. Some of the popular social networking sites are, Facebook, LinkedIn etc.

6. Content communities: Content communities allow users to share online multimedia materials. Some of the popular communities are YouTube, SlideShare, Vimeo, Flickr.

The advantages of social media are: Bring people together, Plan and organise events, Business promotion and scope to express social skills.

Advantages of social media

- Bring people together: Social networking allows people to find long-lost childhood friends and make new ones.
- Plan and organise events: These sites help users to organise and participate in events.
- Business promotion: Social media offers opportunities for businesses to connect with customers, implement marketing campaigns, manage reputation, etc.
- Social skills: These sites allow people to express their views over a particular issue and become an agent for social change.

Limitations in use of social media

- Intrusion to privacy: The personal information of users can be used for illegal activities. Information like the e-mail address, name, location and age can be used to commit online crimes.
- Addiction: Addiction to these sites wastes our valuable time. It will negatively affect our mental states and may lead to depression and tension. It can reduce the productivity of workers in an organisation. Students may lose concentration and this in turn may affect their studies.

- **Spread rumours:** Social media will spread the news very quickly. It can facilitate or worsen a crisis by spreading negative information or misinformation at an incredible speed.

Social media interaction – Best practices

- Avoid unnecessary uploading of personal data like e-mail address, telephone number, address, pictures and videos.
- Setting time schedule for using these sites can save wastage of time.
- In social media websites like wikis and blogs, photo and video sharing are public. What you contribute is available for all to see. Be aware of what you post online. Avoid posting content you may regret later.
- Set your privacy levels in such a way that you know exactly who can see your posts and who can share them. The three basic privacy levels in social media are private, friends and public.

Cyber security

Cyber security refers to protecting computers, programs and data from unauthorised access or destruction. Cyber security is important to protect sensitive business and personal data. Cyber-attack comes in many forms such as virus attack, hacking, phishing etc.

1) Computer Virus: A computer virus is a program that attaches itself to another program or file and spreads from one computer to another. It effects the normal functioning of a computer, such as deleting data, replicating files etc. Most often viruses are spread by e-mail's, portable storage devices etc.

2) Trojan horse: A trojan horse will appear like a useful software but damages data once installed. They can cause serious damages such as deleting files, access confidential data etc. They do not replicate itself.

3) Hacking: A hacker is a person who exploits weakness in a computer system. It is often performed by computer security expert and by computer criminals. Computer expert performs hacking to test security and to find weakness in computer network. Such persons are called 'White hats'. Such hacking is called ethical hacking. Computer criminals who break into computer networks are often called 'Black hats'. They steal important data and destroy files. A

grey hat hacker falls between white and black hackers. Grey hat hacker brings out the weakness in a computer network.

4) **Phishing:** Phishing is an attempt to acquire sensitive information such as user name, password, credit card details. The phishing websites have URLs and home pages similar to their original ones. The act of creating such a misleading website is called spoofing.