



# **ACT - ASSOCIATION OF COMPUTER TEACHERS MALAPPURAM**

## **COMPUTER SCIENCE**

### **CLASS XI**

**Prepared by:**

- ◆ LIJU MATHEW
- ◆ PRIYA M D
- ◆ JESSIE MATHEW
- MARTHOMA HSS CHUNGATHARA
- GHSS PURATHUR
- GHSS VANIYAMBALAM

## Contents

1	The discipline of Computing	3
2	Data Representation and Boolean Algebra	6
3	Components of Computer System	13
4	Principles of programming and problem solving	22
5	Introduction to programming	25
6	Data types and operators	28
7	Control Statements	33
8	Arrays	37
9	String handling using I/O functions	42
10	Functions	44
11	Computer networks	48
12	Internet and mobile computing	56

# CHAPTER 1

## The discipline of Computing

### Computing milestones and machine evolution

#### 1. Development of number systems:

Origin	Base	Features
<i>Egyptians (3000 BC)</i>	10	<ul style="list-style-type: none"><li>• Right to left</li></ul>
<i>Sumerian/ Babylonian</i>	60	<ul style="list-style-type: none"><li>• Sexagesimal system</li><li>• Left to Right</li><li>• Used blank space for 0</li></ul>
<i>Chinese (2500 BC)</i>	10	<ul style="list-style-type: none"><li>• Used bamboo rods to represent digits</li></ul>
<i>India (1500 years ago)</i>		<ul style="list-style-type: none"><li>• Invented a symbol for zero</li><li>• Positional decimal system</li><li>• Hindu-Arabic Numeral system</li></ul>
<i>Greek (500 BC)</i>	10	<ul style="list-style-type: none"><li>• Ionian number system</li></ul>
<i>Roman Numeral</i>		<ul style="list-style-type: none"><li>• 7 letters[ I, V, X, L (50 ),C (100), D(500), M (1000 ) ]</li></ul>
<i>Mayans</i>	20	<ul style="list-style-type: none"><li>• Great accuracy</li></ul>

#### Evolution of the computing machine:

##### 1. Abacus:

- means calculating board.
- Discovered by the Mesopotamians.
- Used for arithmetical calculations.

##### 2. Napier's bones.

- John Napier invented a set of numbered rods to simplify multiplication process (Napier's bones).
- He also invented logarithm.

##### 3. Pascaline:

- Blaise Pascal developed in 1642
- can perform arithmetical operations.
- Operated by dialling a series of wheels, gears and cylinders.

##### 4. Leibniz's calculator :

- Leibniz designed a calculating machine called step reckoner.
- Expanded on Pascal's idea to perform multiplication and division too.

##### 5. Jacquard's loom:

- Joseph Jacquard invented a mechanical loom to simplify a textile manufacturing process.
- The loom controlled by punched cards had the ability to store information.

##### 6. Difference Engine

- Developed by Charles Babbage
- compile mathematical tables, do arithmetical operations and print results automatically

### 7. Analytical Engine

- Developed by Charles Babbage in 1833
- Had many essential features found in modern digital computer.
- Programmable using punched cards
- It had a store(memory) and a separate 'Mill'(Processor)

### 8. Hollerith's machine:

- Herman Hollarith made first electromechanical punched card tabulator with input, output and instructions.
- Used electricity to read, count and sort punched cards.

### 9. Mark-I Computer:

- Developed by Howard Aiken
- Could do all 4 arithmetic operations, logarithmic and trigonometric functions.

## Generations of Computer

### 1) First generation computers:

- Used Vacuum tubes
- The **ENIAC (Electronic Numerical Integrator and Calculator)**, the first general purpose programmable electronic computer(built by J. P. Eckert and John Mauchly.)
- **UNIVAC( UNIVersal Automatic Computer )**, - first commercially successful computer
- **Von Neumann** designed **EDVAC( Electronic Discrete Variable Automatic Computer )** with a memory to store program and data.( **stored program concept**)

### 2) Second Generation Computers :

- Vacuum tubes were replaced by transistors r e d u c i n g s i z e
- less electricity, less expensive.
- Concept of programming language was developed. High Level Languages like FORTAN (FORmula TRANslation), COBOL (COmmon Business Oriented Language) developed.
- Magnetic core memory (Primary memory ) and magnetic disk memory (Secondary memory ).
- The popular computers are **IBM 1401** and **IBM 1620**

### 3) Third Generation Computers:

- **Silicon chips or IC(Integrated Circuits) that contain very small** transistors were developed by Jack Kilby
- Transistors were replaced by IC's.
- It reduced size, increased speed and efficiency, and became cheaper.
- High Level Language BASIC (Beginners All Purpose Symbolic Instruction Code ) was developed .

- *Moore's Law states that the number of transistors on IC's doubles approximately every two years.*
- The popular computers are **IBM 360** and **IBM 370**.

#### 4) Fourth Generation Computers:

- Microprocessors are used (a single chip with Large Scale of Integration (LSI) of electronic components)
- Later LSI circuits were replaced by VLSI ( Very Large Scale Integration ).
- IBM PC and Apple II are popular computers. Programming Languages like. C, C++, Java etc.were developed.

#### e). Fifth Generation Computers:

- They are based on Artificial Intelligence (AI).
- AI is the ability of machines to simulate human intelligence
- Presently in the development stage.
- Common AI programming Languages- LISP and Prolog.

#### Program:

The set of detailed instructions given to a computer for executing specific tasks.

**Programming languages** are artificial languages designed to give instructions to the computers.

- Machine language (Low Level Language -LLL).consists of 0's and 1's. This is the only language understood by the computer.
- Assembly language, with English like words instead of 0's and 1's EDSAC (Electronic Delay Storage Automatic Calculator ) built in 1949 was the first computer to use assembly language.
- Later, High Level Languages (HLL) like, BASIC, C, C++, Java etc were developed.

#### Algorithm and Computer programs:

An algorithm is a step by step procedure to solve a problem.

#### Theory of computing:

- This branch deals with how efficiently problems can be solved based on computation models and related algorithms. The study is based on a mathematical abstraction of computers called model of computation.
- The most commonly used model is Turing machine named after the computer scientist Alan Turing. He is considered as the Father of modern computer science and Artificial intelligence.

#### Turing machine:

- Introduced by Alan Turing
- Theoretical device that uses symbols on a long tape (acting like memory)
- The tape contains cells with a blank, 0 or 1
- The action decided by the current state, symbol currently being read and table of transition rules.
- Considered as the first theoretical development towards the idea of Artificial Intelligence.

## CHAPTER 2

### Data Representation and Boolean Algebra

#### Number system

- The number of symbols used in a number system is called **base** or **radix**.

Number System	Base	Symbols used	Example
Binary	2	0, 1	$(1101)_2$
Octal	8	0, 1, 2, 3, 4, 5, 6, 7	$(236)_8$
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	$(5876)_{10}$
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (A, B, C, D, E, F represents 10, 11, 12, 13, 14, 15 respectively)	$(12AF)_{16}$

- MSD:** The leftmost digit of a number is called **Most Significant Digit (MSD)**.
- LSD:** The right most digit of a number is called **Least Significant Digit (LSD)**.

#### Number Conversions

##### Decimal to binary conversion

Repeated division by 2 and grouping the remainders( 0 or 1)

**Example:** Convert  $(30)_{10}$  to binary.

2	30	Remainders
2	15	0
2	7	1
2	3	1
2	1	1
	0	1

$$(30)_{10} = (11110)_2$$

##### Decimal fraction to binary

- Multiply the decimal fraction by 2.
- Integer part of the answer will be first digit of binary fraction.
- Repeat step 1 and step 2 to obtain the next significant bit of binary fraction.

**Example:** Convert  $(0.625)_{10}$  to binary.

	$0.625 \times 2 = 1.25$
1	$0.25 \times 2 = 0.50$
0	$0.50 \times 2 = 1.00$
1	.00

$$(0.625)_{10} = (0.101)_2$$

### Decimal to Octal conversion

Repeated division by 8 and grouping the remainders.(0,1,2,3,4,5,6 or 7)

**Example:** Convert  $(120)_{10}$  to octal.

8	120	Remainders
8	15	0
8	1	7
8	0	1

$\uparrow$

$(120)_{10} = (170)_8$

### Decimal to Hexadecimal conversion

Repeated division by 16 and grouping the remainders( 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E or F)

**Example:** Convert  $(165)_{10}$  to hexadecimal.

16	165	Remainders
16	10	5
16	0	10 (A)

$\uparrow$

$(165)_{10} = (A5)_{16}$

### Binary to decimal conversion

Multiply binary digit by place value (power of 2) and find their sum.

**Example:** Convert  $(110010)_2$  to decimal.

$$\begin{aligned} (110010)_2 &= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 32 + 16 + 0 + 0 + 2 + 0 = (50)_{10} \end{aligned}$$

Weight	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Bit	1	1	0	0	1	0

### Binary fraction to decimal

Multiply binary digit by place value (negative power of 2) and find their sum.

**Example:** Convert  $(0.101)_2$  to decimal.

$$\begin{aligned} (0.101)_2 &= 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 0.5 + 0 + 0.125 = (0.625)_{10} \end{aligned}$$

Weight	$2^{-1}$	$2^{-2}$	$2^{-3}$
Bit	1	0	1

### Octal to decimal conversion

Multiply octal digit by place value (power of 8) and find their sum.

**Example:** Convert  $(167)_8$  to decimal.

$$\begin{aligned} (167)_8 &= 1 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 \\ &= 64 + 48 + 7 = (119)_{10} \end{aligned}$$

Weight	$8^2$	$8^1$	$8^0$
Octal digit	1	6	7

### Hexadecimal to decimal conversion

Multiply hexadecimal digit by place value (power of 16) and find their sum.

**Example:** Convert  $(2B5)_{16}$  to decimal.

$$\begin{aligned} (2B5)_{16} &= 2 \times 16^2 + 11 \times 16^1 + 5 \times 16^0 \\ &= 512 + 176 + 5 \\ &= (693)_{10} \end{aligned}$$

Weight	$16^2$	$16^1$	$16^0$
Hexadecimal digit	2	B	5

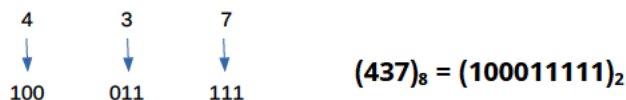
### Octal to binary conversion

Converting each octal digit to its 3 bit binary equivalent.

Octal digit	0	1	2	3	4	5	6	7
Binary equivalent	000	001	010	011	100	101	110	111

**Example:** Convert  $(437)_8$  to binary.

3-bit binary equivalent of each octal digits are



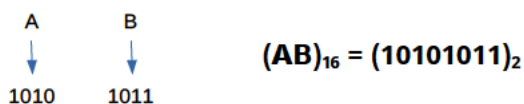
### Hexadecimal to binary conversion

Converting each hexadecimal digit to its 4 bit binary equivalent.

**Example:** Convert  $(AB)_{16}$  to binary.

Octal digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary equivalent	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

4-bit binary equivalent of each hexadecimal digits are

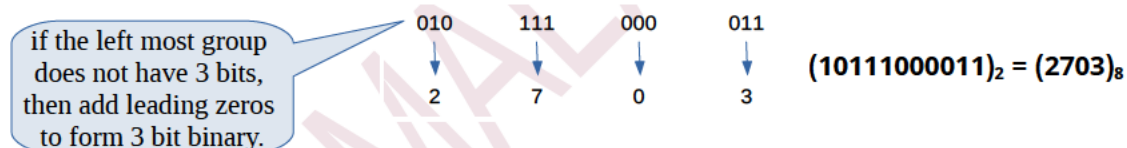


### Binary to octal conversion

Form groups of 3 bits each from right to left and then convert each to its octal form

**Example:** Convert  $(10111000011)_2$  to octal.

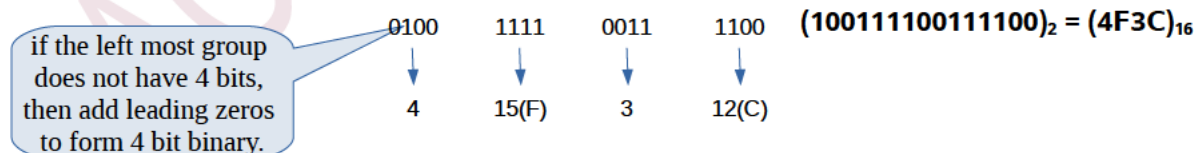
Group the given binary number from right as shown below:



### Binary to Hexadecimal conversion

Form groups of 4 bits each from right to left and then convert each to its hexadecimal form

**Example:** Convert  $(100111100111100)_2$  to hexadecimal.

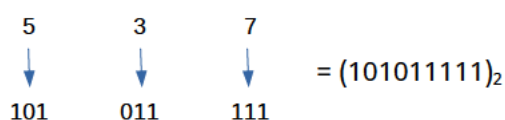


### Octal to hexadecimal conversion

Convert octal to binary and then binary to hexadecimal.

**Example:** Convert  $(537)_8$  to hexadecimal equivalent.

First convert  $(537)_8$  into binary.





Then convert  $(101011111)_2$  into hexadecimal.

$$\begin{array}{ccc} 0001 & 0101 & 1111 \\ \downarrow & \downarrow & \downarrow \\ 1 & 5 & 15(F) \end{array} = (15F)_{16} \quad (537)_8 = (15F)_{16}$$

### Hexadecimal to octal conversion

Convert hexadecimal to binary and then binary to octal.

**Example:** Convert  $(A3B)_{16}$  into octal equivalent.

First convert  $(A3B)_{16}$  into binary.

$$\begin{array}{ccc} A & 3 & B \\ \downarrow & \downarrow & \downarrow \\ 1010 & 0011 & 1011 \end{array} = (101000111011)_2$$

Then convert  $(101000111011)_2$  into octal.

$$\begin{array}{cccc} 101 & 000 & 111 & 011 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 5 & 0 & 7 & 3 \end{array} = (5073)_8 \quad (A3B)_{16} = (5073)_8$$

### Representation of integers

3 methods : (i) Sign and magnitude representation, (ii) 1's complement representation & (iii) 2's complement representation

#### i) Sign and magnitude representation (8 bit form)

- Find binary equivalent of integer
- Make the first bit 1 for negative numbers and 0 for positive numbers.

+23 in sign and magnitude form	
-23 in sign and magnitude form.	

#### ii) 1's complement representation

- If the number is negative, it is represented as 1's complement of 8-bit form binary.
- 1's complement of a binary is obtained by changing 0 to 1 and 1 to 0.
- If the number is positive, the 8-bit form binary itself is the 1's complement.

**Example:** Represent +23 in 1's complement form.

Binary of 23 in 8-bit form =  $(00010111)_2$

+23 in 1's complement form =  $(00010111)_2$  (For +ve numbers, no need to find 1's

complement)

**Example:** Represent -23 in 1's complement form.

Binary of 23 in 8-bit form =  $(00010111)_2$

-23 in 1's complement form =  $(11101000)_2$  (by replacing 0 with 1 and 1 with 0)

### iii) 2's complement representation

- 2's complement of a binary number is calculated by adding 1 to its 1's complement.
- If the number is negative, it is represented as 2's complement of 8-bit form binary.
- If the number is positive, 8-bit form binary itself is the 2's complement.

**Example:** Represent +23 in 2's complement form.

Binary of 23 in 8-bit form =  $(00010111)_2$

+23 in 2's complement form =  $(00010111)_2$  (For +ve numbers, no need to find 2's complement)

**Example:** Represent -23 in 2's complement form.

Binary of 23 in 8-bit form =  $(00010111)_2$

-23 in 1's complement form =  $(11101000)_2$  (by replacing 0 with 1 and 1 with 0)

-23 in 2's complement form =  $11101000 +$

$$\begin{array}{r} 1 \\ \hline = (11101001)_2 \end{array}$$

### Representation of floating point numbers

- Any number in floating point notation contains two parts, **mantissa** and **exponent**.
- Eg: 25.45 can be written as  $0.2545 \times 10^2$ , where 0.2545 is the mantissa and the power 2 is the exponent.

### Representation of characters

Different methods to represent characters in computer memory are: ASCII, Unicode, ISCII, EBCDIC

#### ASCII(American Standard Code for Information Interchange)

- Uses 7 bits per character, can represent only 128 characters.
- ASCII-8, which uses 8 bits, can represent 256 characters.

#### EBCDIC(Extended Binary Coded Decimal Interchange Code)

- 8 bit code used in IBM Machines. It can represent 256 characters.

#### ISCII

- It stands for **Indian Standard Code for Information Interchange** or **Indian Script Code for Information Interchange**.
- It is an 8 bit code for representing various languages of India.

#### Unicode (Universal Code)

- Unicode used 16 bits or more, can represent up to 65,536 characters.
- It can represent almost all written languages of the world.

#### Boolean Algebra: (Algebra of logic)

- It deals with the operations on variables that represent the values 1 and 0.
- The variables which can store logical constants 1 and 0 are called **logical variables or**

**boolean variables.**

**Boolean operations:**

- **OR** (Logical Addition), **AND** (Logical Multiplication) and **NOT** (Logical Negation).

**Logic gate:**

- It is a physical device that can perform logical operations on one or more logical inputs and produce a single logical output. .

**Truth Table:** It is a table that shows Boolean operations and their results.

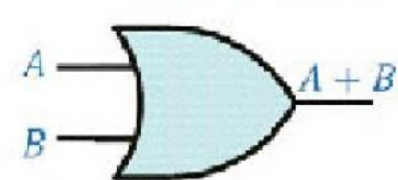
**The OR operator and OR gate.**

Performs logical addition and the symbol used for this operation is + (plus). The expression  $A + B$  is read as A OR B.

**Truth table of OR operation**

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

**OR Gate**

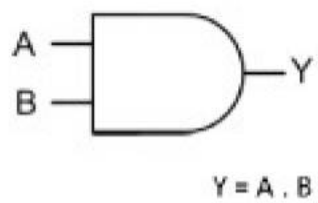


**The AND operator and AND gate**

Performs logical multiplication and the symbol used for this operation is ● (dot). The expression  $A \bullet B$  is read as A AND B.

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

**AND GATE**

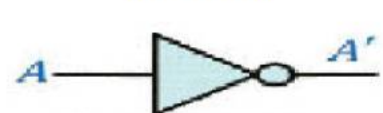


**The NOT operator and NOT gate(Inverter):**

It has only one input and one output. The output is the complement (opposite state) of the input.

A	A'
1	0
0	1

**NOT GATE**



**Basic postulates of Boolean Algebra**

Postulate 1: Principles of 0 and 1

If  $A \neq 0$  then  $A=1$  , if  $A \neq 1$ , then  $A=0$

Postulate 2: OR(Logical Addition)

$$0 + 0 = 0 \quad 0 + 1 = 1 \quad 1 + 0 = 1 \quad 1 + 1 = 1$$

Postulate 3: AND(Logical Multiplication)

$$0.0=0 \quad 0.1=0 \quad 1.0=0 \quad 1.1=1$$

Postulate 4: NOT(Logical Negation or Complement Rule)

$$0'=1 \quad 1'=0$$

**Principle of Duality**

For a Boolean statement, there exists its dual form which can be derived by

1. changing each OR (+) to AND (.)
  2. changing each AND(.) to OR(+)
  3. replace 0 by 1 and 1 by 0.
- eg: Dual of  $A+A'=1$  is  $A.A'=0$

**Basic Theorems of Boolean Algebra**

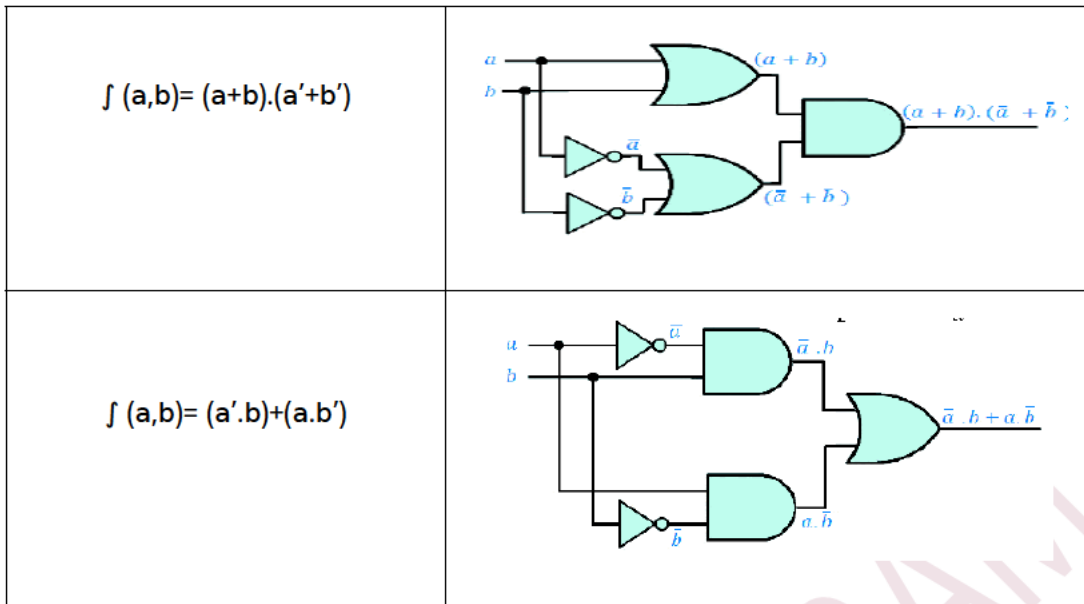
No.	Boolean Law	Statement 1	Statement 2
1	Additive Identity	$0 + X = X$	$1 + X = 1$
2	Multiplicative Identity	$0 . X = 0$	$1 . X = X$
3	Idempotent Law	$X + X = X$	$X . X = X$
4	Involution Law	$\overline{\overline{X}} = X$	
5	Complimentary Law	$X + \overline{X} = 1$	$X . \overline{X} = 0$
6	Commutative Law	$X + Y = Y + X$	$X . Y = Y . X$
7	Associative Law	$X + (Y + Z) = (X + Y) + Z$	$X . (Y . Z) = (X . Y) . Z$
8	Distributive Law	$X . (Y + Z) = X . Y + X . Z$	$X + (Y . Z) = (X+Y) . (X+Z)$
9	Absorption Law	$X + (X . Y) = X$	$X . (X + Y) = X$

**De Morgan's Theorems**

- (i) The complement of sum of Boolean variables is equal to the product of their individual complements .  
 $(X+Y)' = X'.Y'$
- (ii) The complement of product of Boolean variables is equal to the sum of their individual complements .  
 $(X.Y)' = X'+Y'$

**Circuit designs for some Boolean expressions**

Expression	Logical Circuit
$f(A,B) = A'+B$	
$f(X,Y) = X.Y + Y'$	



**Universal Gates:**

**NAND and NOR** (It can implement any Boolean function without using any other gate type)

Gate	Circuit	Logical symbol	Truth table															
<b>NAND</b> (inverted AND gate) eg: For inputs, A and B, the result $Y = (A \cdot B)'$	<p>A circuit diagram of a NAND gate, which is an AND gate followed by an inverter. Inputs are A and B, and the output is <math>Y = (A \cdot B)'</math>.</p>	<p>The standard logical symbol for a NAND gate, consisting of an AND gate symbol with a small circle (bubble) at the output.</p>	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th><math>Y = (A \cdot B)'</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	$Y = (A \cdot B)'$	0	0	1	0	1	1	1	0	1	1	1	0
A	B	$Y = (A \cdot B)'$																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
<b>NOR</b> (inverted OR gate) eg: For inputs, A and B, the result $Y = (A + B)'$	<p>A circuit diagram of a NOR gate, which is an OR gate followed by an inverter. Inputs are A and B, and the output is <math>Y = (A + B)'</math>.</p>	<p>The standard logical symbol for a NOR gate, consisting of an OR gate symbol with a small circle (bubble) at the output.</p>	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th><math>Y = (A + B)'</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	$Y = (A + B)'$	0	0	1	0	1	0	1	0	0	1	1	0
A	B	$Y = (A + B)'$																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

**Implementation of basic gates using NAND and NOR**

Gate	NOT gate	AND gate	OR gate
<b>NAND</b>	<p>A NOT gate implemented using a single NAND gate with both inputs connected to the same input A. The output is <math>(A \cdot A) = A'</math>.</p>	<p>An AND gate implemented using two NAND gates. The first NAND gate takes inputs A and B and produces output <math>(A \cdot B)'</math>. The second NAND gate takes two copies of <math>(A \cdot B)'</math> as input and produces the final output <math>A \cdot B</math>.</p>	<p>An OR gate implemented using three NAND gates. The first two NAND gates take inputs A and B and produce <math>A'</math> and <math>B'</math>. The third NAND gate takes <math>A'</math> and <math>B'</math> as input and produces the final output <math>(A' \cdot B') = A + B</math>.</p>
<b>NOR</b>	<p>A NOT gate implemented using a single NOR gate with both inputs connected to the same input A. The output is <math>(A + A) = A'</math>.</p>	<p>An AND gate implemented using three NOR gates. The first two NOR gates take inputs A and B and produce <math>A'</math> and <math>B'</math>. The third NOR gate takes <math>A'</math> and <math>B'</math> as input and produces the final output <math>(A' + B') = A \cdot B</math>.</p>	<p>An OR gate implemented using two NOR gates. The first NOR gate takes inputs A and B and produces <math>(A + B)'</math>. The second NOR gate takes two copies of <math>(A + B)'</math> as input and produces the final output <math>A + B</math>.</p>

## CHAPTER 3

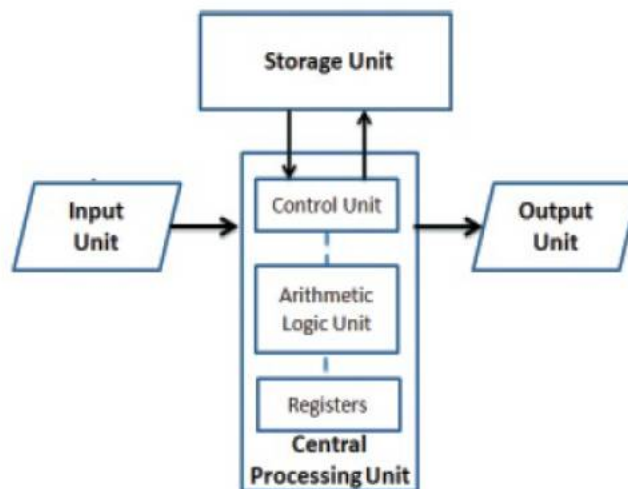
### Components of the Computer System

#### Data processing

- **Data processing** refers to the operations or activities performed on data to generate information.
- **Stages of Data processing**
  1. Capturing data: Data is collected in a proforma(source document)
  2. Input of data: Data collected through source document is fed to computer.
  3. Storage of data: Both data and processed data(information) are stored.
  4. Processing data: Operations like calculations, comparison, sorting, filtering etc are done.
  5. Output of information: The processed data(information) is obtained.
  6. Distribution of information: The information is distributed to users to help them make decisions

#### Functional units of a computer

The functional units of a computer are: Input Unit, Central Processing Unit(CPU), Storage Unit and Output Unit.



#### 1. Input Unit

Accepts instructions and data for processing

Eg:- Keyboard, mouse

#### 2. Central Processing Unit (CPU)

The CPU is the brain of the computer. All major computations and comparisons are made inside the CPU.

The functions of CPU are performed by three components:-

- (i) **Arithmetic Logic Unit(ALU):** It performs all arithmetic and logical operations.
- (ii) **Control Unit(CU):** It manages and co-ordinates all other units of the computer.

(iii) **Registers:** These are the temporary storage areas inside the CPU.

### 3. Storage Unit

Store data and instructions, intermediate results and final results.

### 4. Output Unit

Supplies the results to the outside world.

Eg:- Monitor, Printer

A computer has two major components - *Hardware* and *Software*

## Hardware

- The tangible and visible parts

Eg:- Processor, Motherboard, Keyboard, etc.

### 1. Processor

- *Processor / CPU / Microprocessor* performs all computing and decision making operations and coordinates the working of a computer.

Eg:- Intel core i3, core i5, core i7, AMD Quadcore

Processor has three parts:

- ALU – Arithmetic and Logic Unit ( does Arithmetic and logical calculations )
- CU – Control Unit (controls the operations of all other units)
- Registers (temporary storage areas inside CPU)

Important registers inside a CPU are:

(i) **Accumulator:** stores intermediate arithmetic and logic results.

(ii) **Memory Address Register (MAR):** Stores address of a memory location from which data is read/written

(iii) **Memory Buffer Register (MBR):** It holds the data, to be read/written from memory by processor

(iv) **Instruction Register (IR):** Stores the instructions to be executed by the processor.

(v) **Program Counter (PC):** It holds the address of the next instruction to be executed.

## Motherboard

- A motherboard is a Printed Circuit Board to which all the major components of a computer (processor, memory, etc.) are connected.

## Peripherals and ports

- **Peripherals** are devices connected to a computer system.  
Eg:- input devices, output devices
- **Ports** are slots on the motherboard to connect external devices.

Different ports are Ethernet,USB PS/2, VGA, HDMI, etc.

## Memory

- **Memory** is used to store data and instructions temporarily or permanently.
- Classified into two: **Primary memory** and **Secondary memory**.

### Memory measuring units

Binary Digit =	1 Bit	1 MB (Mega Byte) =	1024 KB
1 Nibble =	4 Bits	1 GB (Giga Byte) =	1024 MB
1 Byte =	8 Bits	1 TB (Tera Byte) =	1024 GB
1 KB (Kilo Byte) =	1024 Bytes	1 PB (Peta Byte) =	1024 TB

#### a. Primary memory

*Primary memory* is used to store data and applications temporarily while the computer is running.

Three types of primary memory are *RAM*, *ROM* and *Cache*.

#### Comparison between RAM and ROM

RAM (Random Access Memory)	ROM (Read Only Memory)
It is faster than ROM.	It is a slower memory.
It stores the operating system, application programs and data temporarily.	Stores program required to boot the computer.
It allows reading and writing.	Allows reading only.
It will lose its data if the power is turned off(volatile)	It keeps data even if the power is turned off.(non volatile)

#### Different types of ROMs are:

- PROM - Programmable ROM
- EPROM - Erasable Programmable ROM
- EEPROM - Electrically Erasable Programmable ROM

#### Cache memory

- Small and fast memory between the processor and RAM.
- Frequently accessed data and instructions are stored in cache memory for quick access.
- Cache is faster and more expensive than RAM.

#### b. Secondary memory (Auxiliary memory)

- Secondary memory is used to store large amount of data or programs permanently.
- It is non-volatile, i.e. its contents are retained even when the device is powered off.
- It is much larger in size , but slower than RAM.
- It acts as a backup.

The major categories of secondary storage devices are:

##### (i) Magnetic storage devices.

Uses plastic tape/metal coated with magnetic material. Eg:- magnetic tapes, hard disks.

##### (ii) Optical storage devices.

Uses laser beam to read/write data. Data is written in continuous spiral form.Eg:- CD, DVD, Blu-Ray

##### (iii) Semiconductor storage devices.



Uses EEPROM chips. Eg:- USB flash drive, Flash memory cards

### Comparison of different types of memory

Storage	Speed	Capacity	Relative Cost	Volatile
Registers	Fastest	Lowest	Highest	Yes
Cache	More Fast	Low	Very High	Yes
RAM	Very Fast	Moderate	High	Yes
Hard Disk	Moderate	Very High	Very Low	No

### Input devices

An **input device** is used to feed data and instructions into the computer.

#### **Different types of input devices are:**

- 1. Keyboard:** It is an input device used to input alphabets, numbers and other characters.
- 2. Mouse:** It is a hand-held device that controls the movement of the cursor on a display screen.
- 3. Light pen:** It is a light-sensitive pointing device used to draw pictures on a computer screen.
- 4. Touch screen:** It allows the user to operate the computer by simply touching on the display screen.
- 5. Graphic tablet:** It allows artists to create graphical images.
- 6. Touchpad:** It is a pointing device used to move the mouse pointer on a display screen.
- 7. Joystick:** It is a pointing device commonly used for playing video games.
- 8. Microphone:** It is used to input sound.
- 9. Scanner:** It is an input device that scans documents such as photos and pages of text.
- 10. Optical Mark Reader (OMR):** It reads pencil marks made on pre-defined positions on the OMR form. It is used to process competitive exam results quickly and accurately.
- 11. Barcode Reader/Quick Response (QR) code reader**
  - Barcode readers are electronic devices for reading printed barcodes.
  - A QR code is similar to barcodes. But barcodes are single dimensional ,QR codes are two dimensional. QR codes store more data like website URLs, plain text, phone numbers, etc.
- 12. Magnetic Ink Character Recognition (MICR) Reader:** It is used in banks for faster clearing of cheques.
- 13. Biometric sensor:** It is used to identify unique human physical features like fingerprints, retina, etc.
- 14. Smart card reader:** A smart card is a plastic card that stores and transacts data (Eg:- ATM Card). Smart card readers are used to access data in a smart card.
- 15. Digital camera:** It can take pictures and videos and that can be transferred to a computer.

### Output devices

*Output device* is used to present information from a computer system.

## **Different types of output devices are:**

### **1. VDU(Visual Display Unit)**

- includes CRT, LC, TFT, LED, Gas plasma, OLED monitors etc
- The information displayed is a soft copy

### **2. LCD projector**

Uses high intensity light beam to display video, images or computer data on a large screen.

### **3. Printer**

Gives hardcopy output.

Classified into two:- **impact** and **non-impact** printers.

**Impact printers** make direct contact with the paper while printing. Eg:- Dot-matrix printers.

#### **Dot Matrix Printer (DMP)**

- Printing material: Ink ribbon
- Cheaper to print, Carbon copy possible, slow, noisy, printer is expensive.

**Non-impact printers** do not touch the paper while printing.

Eg:- Inkjet, Laser and Thermal printers

#### **Inkjet printer**

- Printing material: Liquid ink
- Quiet, high print quality, printers are inexpensive, ink is expensive.

#### **Laser printer**

- Printing material: Ink powder
- Quiet, prints faster, high print quality, Toner is expensive, Device is expensive.

#### **Thermal printer**

- Printing material: Heat sensitive paper.
- Quiet, faster, smaller, lighter, consume less power, portable.

### **4. Plotter**

- Produces hard copies of large graphs and designs on the paper.
- Used in the design of cars, ships, aircrafts, buildings, highways etc.

### **5. Three dimensional (3D) printer**

- Used to print 3D objects.
- It can print ceramic cups, plastic toys, metal machine parts, etc.

### **6. Speakers**

- Output devices that produce sound.

## **e-Waste**

**e-Waste** refers to electronic products nearing the end of their "useful life".

Eg:- discarded computers, mobile phones, television sets, refrigerators, etc.

### **e-Waste disposal methods**

- a. Reuse:** Second-hand use.
- b. Incineration:** Controlled, combustion process in which waste is burned at a high temperature.
- c. Recycling:** Making new products from old devices.
- d. Land filling:** Soil is excavated and e-waste is buried in it.

### **Why should we be concerned about e-Waste?**

- Electronic waste contains some toxic substances such as mercury, lead, etc.
- The toxic materials can cause cancer and many other health problems, if not properly managed.

### **Students' role in e-Waste disposal**

- Stop buying unnecessary electronic equipments.
- Try to repair faulty devices instead of buying a new one.
- Try to recycle electronic equipments.
- Buy rechargeable instead of disposable batteries.
- Buy products with good warranty.

### **Green computing or Green IT**

- The environmentally responsible and eco-friendly use of computers and their resources is known as **green computing**.

#### **To promote green computing the following four approaches are employed:**

- (i) Green design, (ii) Green manufacturing, (iii) Green use, (iv) Green disposal

### **Software**

- Software is a set of programs that help us to use the computer system.
- Two types of software: *System software* and *Application software*

### **System software**

It is a set of one or more programs designed to control the operations of a computer.

### **Components of system software :**

Operating system, Language processors and Utility software.

#### **a. Operating system**

- Set of programs that acts as an interface between the user and computer hardware. Example:- DOS, Windows, Unix, Linux

#### **Major functions of an Operating System are:**

- Process management, Memory management, File management and Device management

#### **b. Language processor**

- It translates programs written in high level language or assembly language into machine language.

## Computer languages

**Low Level Language:** machine-oriented languages. Two types:

*a. Machine language:* Uses only binary digits 0 and 1.

*b. Assembly language:* is an intermediate-level symbolic programming language. It uses mnemonic symbols like ADD, SUB, etc.

**High Level Language** is like English language and is simpler to understand. Not understandable to the computer. Example: C++

### Types of language processors

**Assembler:** Converts assembly language into machine language.

**Interpreter:** Converts a high level language program into machine language line by line.

**Compiler:** Converts the whole high level language program into machine language at a time.

### c. Utility software

- Set of programs which help in system maintenance tasks.

#### Some of the utility programs are:

*(i) Compression tools:* Large files can be compressed so that they take less storage area.

Eg: WinZip, WinRAR

*(ii) Disk defragmenter:* Rearranges fragmented files on a computer hard disk.

*(iii) Backup software:* Duplicates disk information so as to use in the event of a system crash.

*(iv) Antivirus software:* Scans the computer system for viruses and removes them.

Eg: Norton Antivirus, Kaspersky

### Application software

- Software developed for specific application is called application software.
- It includes *general purpose software* and *specific purpose software*.

#### a. General purpose software

- Software that can be used for a variety of tasks. It is classified as:
  - Word processing software                      Eg: MS Word, Open Office Writer
  - Spreadsheet software                              Eg: Microsoft Excel, Open Office Calc
  - Presentation software                              Eg: Microsoft PowerPoint, Open Office Impress.
  - Database software                                      Eg: Microsoft Access, Oracle, MySQL
  - Multimedia software                                  Eg: Adobe Flash, Media Player

#### b. Specific purpose software

- Tailor-made software to satisfy the needs of an organisation. Also known as *customised software*.

Eg: Payroll System, Inventory Management System, HR Management System.

### Free and open source software

- It gives the user the freedom to use, copy, distribute, examine, change and improve the software.  
Eg:- GNU/Linux, GIMP, Mozilla Firefox, OpenOffice.org

### Four Freedoms of Free Software

Freedom 0 - The freedom to run program for any purpose.

Freedom 1 - The freedom to study how the program works and adapt it to your needs.

Freedom 2 - The freedom to distribute copies of the software.

Freedom 3 - The freedom to improve the program and release your improvements to the public.

### Freeware and Shareware

Freeware	Shareware
Copyrighted software available free of cost for an unlimited period.	Commercial software give users a chance to try the software before buying it.
All the features are free.	All features are not available.
Freeware programs can be distributed free of cost.	Shareware may or may not be distributed freely.

### Proprietary software

- It is an exclusive property of its developer and cannot be copied or distributed without permission.  
Eg:- Microsoft Windows OS, MS Office

### Humanware

- Humanware or liveware refers to humans who use the computer.  
Eg:- Computer programmers, Computer Operators, System Administrators, etc.

## CHAPTER 4

# Principles of Programming and Problem Solving

### Approaches in problem solving

- Top down design: A complex problem is broken down into different tasks and each task is solved.
- Bottom up design: Solution for main module will be developed only after solving the sub modules.

### Phases in programming

- Problem identification
- Preparing Algorithms and Flowcharts
- Coding the program
- Translation
- Debugging
- Execution and testing
- Documentation

### Problem Identification

- In this phase we will be able to identify the data involved in processing, its type and quantity, formula to be used, activities involved, and the output to be obtained.

### Algorithm

- It is a step-by-step procedure to solve a problem.


#### Characteristics of an Algorithm

- Should begin with instructions to accept inputs.
- Use variables to refer the data
- Each and every instruction should be precise and unambiguous.
- The total time to carry out all the steps in the algorithm must be finite.
- After performing the instructions given in the algorithm, the desired results must be obtained.




### Flowcharts

- The pictorial representation of an algorithm is known as **flowchart**.

#### Flowchart symbols

1.  **Ellipse:** used to indicate START and STOP.

2.  **Parallelogram:** used as the input/output symbol.

3.  **Rectangle:** used to represent the processing step.
4.  **Rhombus:** used as decision symbol.
5.  **Flow lines:** used to indicate the flow of operation.

#### Advantages of flowchart

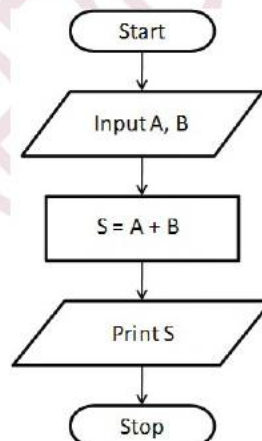
- Better communication
- Effective analysis
- Effective synthesis
- Effective synthesis

#### Limitations of Flowchart

- Time consuming
- Modification is difficult

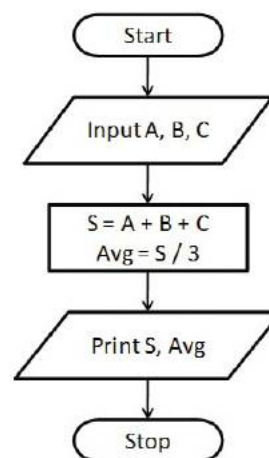
#### Eg: Write an algorithm and draw a flowchart to find the sum of two numbers

- Step 1: Start
- Step 2: Input A, B
- Step 3:  $S = A + B$
- Step 4: Print S
- Step 5: Stop



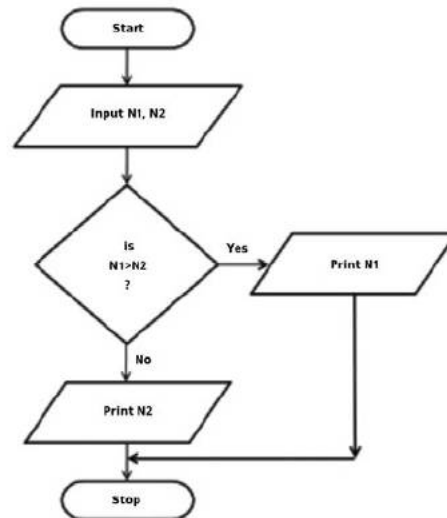
#### Eg: Write an algorithm and draw a flowchart to find the sum and average of three numbers

- Step 1: Start
- Step 2: Input A, B, C
- Step 3:  $S = A + B + C$
- Step 4:  $Avg = S / 3$
- Step 5: Print S, Avg
- Step 6: Stop



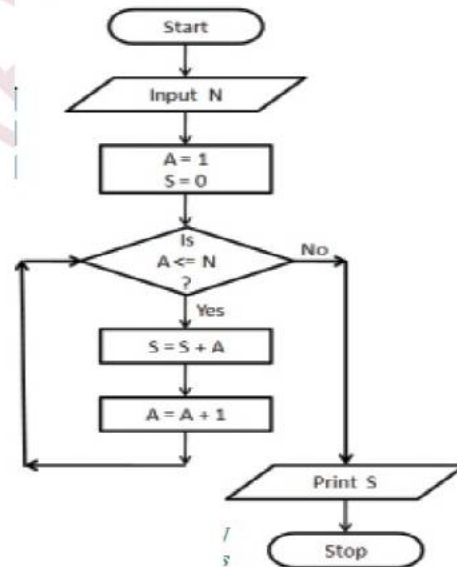
**Eg: Write an algorithm and draw a flowchart to find the biggest of two numbers.**

- Step 1: Start
- Step 2: Input N1, N2
- Step 3: If  $N1 > N2$  Then
- Step 4: Print N1
- Step 5: Else
- Step 6: Print N2
- Step 7: End of If
- Step 8: Stop



**Eg: Write an algorithm and draw a flowchart to print the sum of the first N natural numbers**

- Step 1: Start
- Step 2: Input N
- Step 3:  $A = 1, S = 0$
- Step 4: Repeat Steps 5 and 6 While  $(A \leq N)$
- Step 5:  $S = S + A$
- Step 6:  $A = A + 1$
- Step 7: Print S
- Step 8: Stop



### Coding

- The process of writing program instructions to solve a problem using a High Level Language.

### Translation

- Process of converting a program written in high level language(**source code**) into its equivalent version in machine language(**object code**)





## Debugging

- Programming errors are known as '**bugs**'. The process of detecting and correcting errors is called **debugging**.

### There are three types of errors:

**Syntax errors:** Errors which occur when the rules or syntax of the programming language are not followed.

**Logical errors:** Logical error is due to improper planning of the program logic.

**Run-time errors:** Errors which occur during program execution. Eg:- division by zero

## Execution and testing

- Running the program to process the test data that will produce 'known results'.

## Documentation

- Writing comments in the source code is known as **internal documentation**.
- It will help in debugging process and program modification at later stage.
- Preparing user manual and system manual are known as **external documentation**.

## CHAPTER 5

### Introduction to Programming

#### Character set

- Fundamental units(set of valid characters) of C++ Language. It consists of letters, digits, special characters, white spaces

#### Tokens

- Fundamental building blocks of C++ program. (Lexical units)

#### Classification of Tokens

- **Keyword (Reserved word):** They convey a specific meaning to the compiler.  
Eg: float, if, break, switch
- **Identifiers:** Names given to different program elements.
  - (a) *Variable:* Name given to memory location.
  - (b) *Label:* Name given to a statement.
  - (b) *Function name:* Name given to a group of statements.

#### Rules to form an identifier:

- Identifier is a sequence of letters , digits and underscore.
- The first character must be letter or underscore

- Keywords cannot be used as identifiers.
- Special characters or white spaces cannot be used  
Eg: score, score1, score\_1, INT

- **Literals(Constants):** Constant values used in program.

(a) Integer literals: Whole numbers. Eg: 23, -145

(b) Floating literals: Constants having fractional parts. Eg: 12.5, 1.87E05

(c) Character literals: A character in single quotes. Eg: 'a', '8'

Escape sequences are character constants used to represent non graphic symbols. Eg: '\n,' '\t'

(d) String literals: One or more characters within double quotes. Eg: "a", "score1"

- **Operators:** Symbols that indicate an operation. Eg: +, <, \*, &&

- **Punctuators:** Special symbols used in C++ program. Eg: # ; ( ] }

<b>IDE-</b> Integrated Development Environment
--

In Geany, C++ programs are saved with the extension <b>.cpp</b>
---

## CHAPTER 6

### Data Types and Operators

**Data types:** Used to identify nature and type of data stored in a variable.

**Fundamental data types:**

Data type	Type of value	Memory (bytes)	Eg:
<b>void</b>	null or empty data	0	
<b>char</b>	character values	1	'A', '\n'
<b>int</b>	integer values	4	84, -4
<b>float</b>	real values	4	5.6, -89.5
<b>double</b>	Real values(more precision than float)	8	

} Integral datatypes

} Numerical datatypes

} Floating point datatypes

Float data has a precision of 7 digits and double data has a precision of 15 digits

**Type modifiers**

- Used to change the size, range or precision of a data type
- ***Type modifiers in C++ are long, short, signed and unsigned***

**Variables:** Names given to memory locations.

Eg: int Num=18;

- |      |  |
|------|--|
| 1001 | i. <b>Variable name</b> : The name of the variable. ( <b>Num</b> )           |
| 18   | ii. <b>Memory address(L-value)</b> :The memory address.( <b>1001</b> )       |
| Num  | iii. <b>Content(R-value)</b> :The value stored in the variable.( <b>18</b> ) |

**Operators:** Symbols that indicate an operation. Eg: +, <, \*, &&

*In the expression a+b, + is the **operator** and a and b are the **operands**.*

**Classification of operators**

**1. Based on number of operands:**

Category	No. of operands	Example
Unary	1	Unary +, unary -, ++,--
Binary	2	+, &&, <
Ternary	3	?:

## 2. Based on nature of operation

(a) **Arithmetic operators** : Used for arithmetic operations such as addition(+), subtraction(-), multiplication(\*), division(/) and modulus (%).

x	y	x+y	x-y	x*y	x/y	x%y
7	3	10	4	21	2	1(remainder)

(b) **Relational operators**:

Used for comparing numeric data, resulting in either **true** (1) or **false** (0). The relational operators are <(less than), >(greater than), <=(less than or equal to), >=(greater than or equal to), ==(equal to), !=(not equal to).

x	y	x<y	x>y	x<=y	x>=y	x==y	x!=y
7	3	0	1	0	1	0	1

(c) **Logical operators**:

Used to combine logical values, resulting in true or false. The logical operators are &&(AND), || (OR) and !(NOT)

E1	E2	E1&&E2	E1  E2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

E	!(E)
0	1
1	0

Eg: (2>5) && (4<6) results in 0

(2>5) || (4<6) results in 1

!(2>5) results in 1

**Input / Output operators:**

- The **get from (extraction)** operator (>>) is used for input. Eg: cin>>a;
- The **put to (insertion)** operator (<<) is used for output. Eg: cout<<a;

**Assignment operator (=)**

- Used to store a value in a variable. Eg: a=5;

**Difference between = and == operators.**

=	==
Assignment operator	Relational operator
Assigns value to a variable	Compares values

**Arithmetic assignment operators (C++ short hands)**

The operators which combine arithmetic and assignment operators. They are +=, -=, \*=, /=, %=

eg: a=a+5 can be given as a+=5      a=a/10 can be given as a/=10

## Increment (++) and Decrement (--) operators

++ is used to **increment** the value in a variable by 1.

-- is used to **decrement** the value in a variable by 1.

**a++** (post increment form) and **++a** (pre increment form) are same as  $a=a+1$  or  $a+=1$

**a--** (post decrement form) and **--a** (pre decrement form) are same as  $a=a-1$  or  $a-=1$

(A post-form denotes use, then change method and a pre-form denotes change, then use method)

<pre>int m=5; n=m++;</pre> <p>(Now n -&gt; 5 and m -&gt; 6)</p>	<pre>int m=5; n=--m;</pre> <p>(Now n -&gt; 4 and m -&gt; 4)</p>
---	---

## sizeof operator

It is a unary compile time operator that returns the amount of memory space(in bytes) allocated for the operand. Eg: `sizeof(int)` --> *gives the value 4*, `sizeof(3.2)` --> *gives the value 8*

## Precedence of operators

It is the order in which the operations are performed in an expression. The order of execution is ( ), ++, --, !, Unary +, Unary -, sizeof, \*, /, %, +, - etc

**Expressions:** Combination of operators and operands.

(a) Arithmetic expressions: contains arithmetic operators. Eg:  $m + n * y$

(i) **Integer expressions:** contains integer operands, Result is integer.

(ii) **Floating point(Real) expressions:** contains real operands. Result is float value.

(iii) **Constant expressions:** contains constant values. (eg:  $5+7/3$ )

(b) Relational expressions: contains relational operators. Result is either true or false. Eg:  $x > y$

(c) Logical expressions: contains logical operators. Result is either true or false. Eg:  $a < b \ \&\& \ c > d$

## Type conversion

- Conversion of the data type of an operand to another.

### Two types of conversion:

**1. Implicit (Type promotion):** This is done by the compiler and the conversion is from lower data\_type to higher.

Eg:  $5/2.0 \Rightarrow 2.5$  (Here, int datatype of 5 is converted to float by compiler. Thus the result of the float expression is also float)

**2. Explicit (Type casting):** This is done by the programmer explicitly and conversion can be to any data\_type.

Eg:  $5/(int)2.0 \Rightarrow 2$  (Here, programmer uses type casting (int) to convert the float data type of 2.0 to int. Thus the result of this integer expression is also an integer.)

**Statements:** Smallest executable unit of a program. *C++ statement ends with semicolon (;)*

**1. Declaration statement:** Specifies the type of data that will be stored in a variable.

Syntax: datatype var\_name;

eg: float avg;

### Variable Initialisation:

(a) Supplying the initial value to a variable at the time of declaration.

eg: int n =5 ; or int n(5);

(b) Supplying the initial value to a variable during execution (dynamic initialisation).

eg: float sum=a+b;

### const - The Access Modifier

The keyword const is used to create symbolic constants whose value can never be changed during program execution.

Eg: const float pi = 3.14;

**2. Assignment statements:** It stores a value to a variable. Eg: a=15;

**3. Input statements:** Specifies an input operation. Eg: cin>>score;

**4. Output statement:** Specifies output operation. Eg: cout<<score;

The multiple use of input/output operators in a single statement is called **cascading of I/O operators**.

Eg: cin>>x>>y>>z;  
cout<<"Sum="<<S;

### Structure of a C++ Program

```
#include <headerfile>      ----> line 1
using namespace identifier; ----> line 2
int main()                  ----> line 3
{
    statements;
    return 0;
}
```

**Line 1:** Preprocessor directive :

Instructs the compiler to perform an action before actual compilation. Starts with the symbol #.

eg: #include<iostream> --instruction to link the header file iostream.

**Line 2:** namespace statement:

Tells the compiler to use namespace **std** in this program. std is the standard namespace in which a lot of objects, including cin and cout are defined.

**Line 3:** Function header:

main() is the essential function for every C++ program. A C++ program execution starts and ends within the main() function.

### A sample program

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Hello, Welcome to C++";
    return 0;
}
```

### Guidelines for coding

- Use suitable naming convention for identifiers
- Use clear and simple expressions
- Use comments wherever needed
- Give proper indentation

### Comments

Lines in code that are added to describe the program. They provide for internal documentation.

There are two ways to write comments in C++:

- *Single line comment:* The characters `//` (two slashes) is used to write single line comments.
- *Multiline comments:* Anything written within `/*` and `*/` is treated as comment

### Write a C++ program to input two numbers and find their sum.

```
#include <iostream>
using namespace std;
int main()
{
    int num1, num2, sum;
    cout<<"Enter two numbers: ";
    cin>>num1>>num2;
    sum=num1+num2;
    cout<<"Sum of the entered numbers = "<<sum;
    return 0;
}
```



# CHAPTER 7

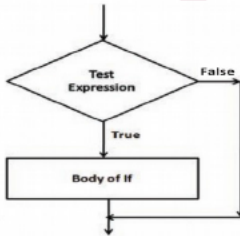
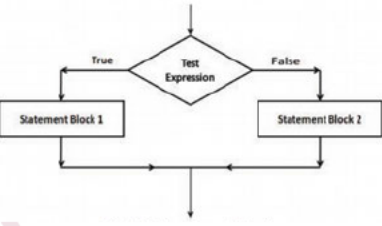
## Control Statements

### Control Statements

The statements used to change the normal sequential flow of program execution.

Two types: (1) Decision (2) Iteration

**1. Decision (Selection) Statements** : Statements are selected for execution based on a condition.

Statement	Syntax	Example
if statement	<pre>if ( test expression ) {     statement block; }</pre> 	<pre>if (mark&gt;=18) cout&lt;&lt;"Passed" ;</pre>
if ..... else statement	<pre>if ( test expression ) {     statement block 1; } else {     statement block 2; }</pre> 	<pre>if (mark&gt;=18)     cout&lt;&lt;"Passed" ; else     cout &lt;&lt;"Failed " ;</pre>
if .... else ... if ladder	<pre>if ( test expression ) {     statement block 1; } else if ( test expression 2 ) {     statement block 2 ; } else if (test expression 3 ) {     statement block 3 ; } ..... else {     statement block n; }</pre>	<pre>if (mark&gt;=80)     cout&lt;&lt;"A Grade"; else if(mark&gt;=60)     cout&lt;&lt;"B Grade " ; else if(mark&gt;=40)     cout&lt;&lt;"C Grade"; else     cout&lt;&lt;"D Grade";</pre>
switch statement	switch(variable / expression)	switch(n)

<pre> { case constant_1 : statement 1;                 break; case constant_2 : statement 2;                 break; case constant_3 : statement 3;                 break; ..... default : statement n; } </pre>	<pre> { case 1 : cout&lt;&lt;"One";         break; case 0 : cout&lt;&lt;"Zero"         break; default :         cout&lt;&lt;"Invalid"; } </pre>
---	---

### Comparison between switch and if else if ladder

switch	if else if ladder
Permits multiple branching.	Permits multiple branching.
Can be used only for checking equality	Any relational or logical condition can be checked
Case constant is integer or character	Can compare against a set of values including floating numbers
Uses <b>default</b> when no match is found, and <b>break</b> is used to exit.	When no condition is true, else block executes.

***All switch statements can be converted to else if ladder. But not all if-else if can be converted to switch.***

### Conditional Operator ( ?: )

It is a ternary operator in C++ . It is an alternative to if ..... else statement

**Syntax : (test expression) ? True statement : false statement ;**

Eg : (mark>=18) ? cout<<"Passed": cout<<"Failed" ;

***is same as***

```

if (mark>=18) cout<<"Passed";
else cout<<"Failed" ;

```

### 2. Iteration (Looping) statements

Statements that allow repeated execution of a set of one or more statements.

***There are 3 loops in C++*** : while, for, do-while

A loop has the following components:

- Loop components:**
- (1) initialisation: statement that gives starting value to loop variable(**i=1**)
  - (2) condition - the test expression. (**i<=10**)
  - (3) updation - statement that changes the value in loop variable (**i++**)
  - (4) body of loop- set of statements that execute repeatedly (**cout<<i;**)

#### for loop

```

for (i=1;i<=10;i++)
cout << i;

```

#### while loop

```

i=1;
while(i<=10)
{

```

#### do while loop

```

i=1;
do
{

```

```

        cout<<i;
        i++;
    }
    cout<<i;
    i++;
}while(i<=10);

```

Here, *i* is the loop variable ( its value controls the loop).

[All the three loops given above produce the same output – displays numbers from 1 to 10]

Statement	Syntax	Example
for loop	for (initialisation ; test expression ; update expression) { Body of the loop; }	for (i=1;i<=10;++i) cout<<i;
while loop	Initialisation; while (test expression) { Body of the loop; Update expression ; }	i=1 ; while (i<=10) { cout<<i; ++i; }
do....while loop	Initialisation; do { Body of the loop; Update expression ; }while(test expression);	i=1 ; do { cout<<i; ++i; }while(i<=10);

### Entry controlled loop

1. condition before loop body
  2. loop will run only if condition is true
- eg: for, while

### Exit controlled loop

1. condition after loop body
  2. loop will run at least once even if condition is true/false
- eg: do while

### Nesting of loops

- Placing a loop inside the body of another loop

```

for( i=1; i<=2; ++i)
{
    for(j=1; j<=3; ++j)
    {
        cout<< "\t" << i << " " << j;
    }
}

```

Output: 1 1 1 2 1 2 2 3

*Any control statement can be nested with another control statement.*

## Jump Statements

Jump statements are used to transfer the program control from one place to another.

C++ provides four jump statements

1. **goto** - transfers the program control to a label, which is an identifier.

Syntax :

```
goto label;
.....;
.....;
label: .....;
.....;
```

2. **break** - transfers the program control outside the loop or switch.

```
Eg: for (i=1;i<=10;i++)
{ if (i==5) break;
  else cout<<i<<'\t'; }
```

**Output:** 1    2    3    4

3. **continue**- transfers the program control to the next iteration

```
Eg: for (i=1;i<=10;i++)
{ if (i==5) continue;
  else cout<<i<<'\t'; }
```

**Output:** 1    2    3    4    6    7    8    9    10

4. **return** - used to transfer control back to the calling program or to come out of a function.

<b>break statement</b>	<b>continue statement</b>
<ul style="list-style-type: none"><li>• Used with switch and loops.</li><li>• Brings the program control outside the switch or loop by skipping the rest of the statements within the block.</li><li>• Program control goes out of the loop even though the test expression is True.</li></ul>	<ul style="list-style-type: none"><li>• Used only with loops.</li><li>• Brings the program control to the beginning of the loop by skipping the rest of the statements within the block.</li><li>• Program control goes out of the loop only when the test expression becomes False.</li></ul>

`exit()` is a built in function in C++ which terminates the program (header file- `cstdlib`)

## CHAPTER 8

### ARRAYS

#### Array

Array is a collection of elements of same type placed in contiguous memory location

Eg : A[10] , NUM[20]

Each element in an array can be accessed using its position called index number or subscript.

An array index starts from 0. The elements of an array with ten elements are numbered from 0 to 9.

#### Array declarations

data\_type array name[size]; where size is the number of memory locations in the array.

Eg: int N[10];

float A[5];

char name[25];

#### Array initialization

Giving values to the array elements at the time of array declaration is known as array initialization.

Eg : int N[10]={12,25,30,14,16,18,24,22,20,28};

float A[5]={10.5,12.0,5.75,2.0,14.5};

char word[7]={'V', 'I', 'B', 'G', 'Y', 'O', 'R'};

#### Memory allocation for arrays

total bytes = size of data\_type x size of the array

Eg:

The number of bytes needed to store the array int A[10] is  $4 \times 10 = 40$  bytes.

The number of bytes needed to store the array float P[5] is  $4 \times 5 = 20$  bytes.

The memory needed to store the array char S[25] is  $1 \times 25 = 25$  bytes.

The memory needed to store the array int Num[ ]={25,65,14,24,27,36} is  $4 \times 6 = 24$  bytes.

#### Array traversal operation

Accessing each element of an array at least once in a program is called array traversal.

- **C++ statement to display the elements of an array N[10]**

```
for(i=0 ; i<10 ; ++i)
    cout<<N[i];
```

- **C++ statement to input 10 numbers into an array N[10]**

```
for(i=0 ; i<10 ; ++i)
    cin>>N[i];
```

#### Array operations :

##### 1. Traversal :

- Accessing each element of an array at least once to perform any operation is known as **traversal** operation.
- Eg. Displaying all the elements of an array

#### Write a C++ program to input 10 numbers into an array and display the sum of numbers

```
#include<iostream>
using namespace std;
int main()
[
int N[10], i , s=0 ;
cout<<"Enter 10 numbers ";
for(i=0 ; i<10 ; ++i)
{
```

```

        cin>>N[i];
        s=s+N[i];
    }
    cout<<"Sum of 10 numbers is "<<s;
    return 0;
}

```

**(2). Sorting:**

- Arranging elements of an array in ascending or descending order.

**(a). Selection sort:**

- The array is divided into sorted part and unsorted parts.
- To arrange in ascending order, first find the smallest element and exchange it with the first element.
- Then find the second lowest element and exchange the element in second position.
- This process is repeated till the array is sorted.

The process of finding the smallest element and exchanging it with the element in respective position is known as a '**PASS**'.

**(b). Bubble sort:**

- This repeatedly steps through the list, compares each pair of adjacent items and swaps them if they are in the wrong order.
- This passing procedure repeats until no swaps are required, indicating that the list is sorted.

<b>C++ code to input N elements into an array AR and sort in ascending order</b>	
<b>Selection sort</b>	<b>Bubble sort</b>
<pre> for(l=0; l&lt;N; l++) cin&gt;&gt;AR[l]; for(l=0; l &lt; N-1; i++) { MIN=AR[l]; POS=l; for(j = l+1; j &lt; N; j++) if(AR[j]&lt;MIN) { MIN=AR[j]; POS=j; } if(POS != l) { AR[POS]=AR[l]; AR[l]=MIN; } } cout&lt;&lt;"Sorted array is: "; for(l=0; l&lt;N; l++) cout&lt;&lt;AR[l]&lt;&lt;"\t"; </pre>	<pre> for(l=0; l&lt;N; l++) cin&gt;&gt;AR[l]; for(l=1; l&lt;N; l++) for(j=0; j&lt;N-l; j++) if(AR[j] &gt; AR[j+1]) { TEMP = AR[j]; AR[j] = AR[j+1]; AR[j+1] = TEMP; } cout&lt;&lt;"Sorted array is: "; for(l=0; l&lt;N; l++) cout&lt;&lt;AR[l]&lt;&lt;"\t"; </pre>

**(3). Searching:**

- Process of finding the location of the given element in an array.

**(a) Linear Search(Sequential Research)**

- Linear search or sequential search is a method for finding a particular value in a list.
- Linear search consists of checking each element in the list, starting from the first element until

the desired one is found(successful) or the end of list is reached(unsuccessful).

### (a) Binary Search

- This search is performed on a sorted array.
- It checks the middle element, eliminates half of the list from consideration and proceeds with the other half.
- If the middle element is the searched value, position is found, otherwise upper half or lower half is chosen for search based on whether the searched value is greater or less than the middle element.

C++ code to input N elements into an array AR and search for an element input as ITEM	
Linear Search	Binary Search
<pre>for(l=0; l&lt;n; l++) cin&gt;&gt;AR[l]; cout&lt;&lt;"Enter the item you are searching for: "; cin&gt;&gt;ITEM; for(l=0; l&lt;N; l++) if(AR[l] == ITEM) { LOC=l; break; } if(LOC!=-1) cout&lt;&lt;"The item is found at position "&lt;&lt;LOC+1; else cout&lt;&lt;"The item is not found in the array";</pre>	<pre>for(l=0; l&lt;N; l++) cin&gt;&gt;AR[l]; cout&lt;&lt;"Enter the item to be searched: "; cin&gt;&gt;ITEM; FIRST=0; LAST=N-1; while(FIRST&lt;=LAST) { MIDDLE=(FIRST+LAST)/2; if(ITEM == AR[MIDDLE]) { LOC = MIDDLE; break; } if(ITEM &lt;AR[MIDDLE]) LAST = MIDDLE-1; else FIRST = MIDDLE+1; } if(LOC != -1) cout&lt;&lt;"The item is found at position "&lt;&lt;LOC+1; else cout&lt;&lt;"The item is not found in the array";</pre>

### Comparison of linear and binary search methods

Linear search method	Binary search method
<ul style="list-style-type: none"> <li>• The elements need not be in any order</li> <li>• Takes more time for the process</li> <li>• May need to visit all the elements</li> <li>• Suitable when the array is small</li> </ul>	<ul style="list-style-type: none"> <li>• The elements should be in sorted order</li> <li>• Takes very less time for the process</li> <li>• All the elements are never visited</li> <li>• Suitable when the array is large</li> </ul>

### Two Dimensional(2D) Arrays

It is an array in which each element itself is an array.

**AR[m][n]**

- is a 2D array, which contains m single dimensional arrays, each of which has n elements.

- is a table containing m rows and n columns.

**Syntax to declare a 2D array :**

data\_type array\_name[rows][columns];

**Memory** = sizeof(base type) × number of rows × number of columns

eg: int marks[5][4] requires 4×5×4=80 bytes.

**C++ code to create a matrix with m rows and n columns**

```
#include <iostream>
using namespace std;
int main()
{ int m, n, row, col, mat[10][10];
cout<< "Enter the order of matrix: ";
cin>> m >> n;
cout<<"Enter the elements of matrix\n";
for (row=0; row<m; row++)
for (col=0; col<n; col++)
cin>>mat[row][col];
cout<<"The given matrix is:";
for (row=0; row<m; row++)
{
cout<<endl;
for (col=0; col<n; col++)
cout<<mat[row][col]<<"\t";
}
return 0;
}
```

To accept elements

To display elements  
in  
matrix format

<b>To find the sum of two conformable matrices</b>	<b>To find the row sum and column sum of a matrix</b>
<pre>#include &lt;iostream&gt; using namespace std; int main() { int m1, n1, m2, n2, row, col; int A[10][10], B[10][10], C[10][10]; cout&lt;&lt;"Enter the elements of first matrix\n"; for (row=0; row&lt;m1; row++) for (col=0; col&lt;n1; col++) cin&gt;&gt;A[row][col]; cout&lt;&lt;"Enter the elements of second matrix\n"; for (row=0; row&lt;m2; row++) for (col=0; col&lt;n2; col++) cin&gt;&gt;B[row][col]; for (row=0; row&lt;m1; row++) for (col=0; col&lt;n1; col++) C[row][col] = A[row][col] + B[row][col]; cout&lt;&lt;"Sum of the matrices:\n"; for(row=0; row&lt;m1; row++) { cout&lt;&lt;endl; for (col=0; col&lt;n1; col++) cout&lt;&lt;C[row][col]&lt;&lt;"\t"; } }</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main() { int ar[10][10], rsum[10]={0}, csum[10]={0}; int m, n, row, col; cout&lt;&lt;"Enter the number of rows &amp; columns in the array: "; cin&gt;&gt;m&gt;&gt;n; cout&lt;&lt;"Enter the elements\n"; for(row=0; row&lt;m; row++) for(col=0; col&lt;n; col++) cin&gt;&gt;ar[row][col]; for(row=0; row&lt;m; row++) for(col=0; col&lt;n; col++) { rsum[row] += ar[row][col]; csum[col] += ar[row][col]; } cout&lt;&lt;"Row sum of the 2D array is\n"; for(row=0; row&lt;m; row++) cout&lt;&lt;rsum[row]&lt;&lt;"\t"; }</pre>



<pre>return 0; }</pre>	<pre>cout&lt;&lt;"\nColumn sum of the 2D array is\n"; for(col=0; col&lt;n; col++) cout&lt;&lt;csum[col]&lt;&lt;"\t"; return 0; }</pre>
<p><b><u>To find the transpose of a matrix</u></b></p> <pre>#include &lt;iostream&gt; using namespace std; int main() { int ar[10][10], m, n, row, col; cout&lt;&lt;"Enter the order of matrix: "; cin&gt;&gt;m&gt;&gt;n; cout&lt;&lt;"Enter the elements\n"; for(row=0; row&lt;m; row++) for(col=0; col&lt;n; col++) cin&gt;&gt;ar[row][col]; cout&lt;&lt;"Original matrix is\n"; for(row=0; row&lt;m; row++) { cout&lt;&lt;"\n"; for(col=0; col&lt;n; col++) cout&lt;&lt;ar[row][col]&lt;&lt;"\t"; } cout&lt;&lt;"\nTranspose of the entered matrix is\n"; for(row=0; row&lt;n; row++) { cout&lt;&lt;"\n"; for(col=0; col&lt;m; col++) cout&lt;&lt;ar[col][row]&lt;&lt;"\t"; } return 0; }</pre>	<p><b><u>To find the sum of major diagonal elements of a matrix</u></b></p> <pre>#include &lt;iostream&gt; using namespace std; int main() { int mat[10][10], n, i, j, s=0; cout&lt;&lt;"Enter the rows/columns of square matrix: "; cin&gt;&gt;n; cout&lt;&lt;"Enter the elements\n"; for(i=0; i&lt;n; i++) for(j=0; j&lt;n; j++) cin&gt;&gt;mat[i][j]; cout&lt;&lt;"Major diagonal elements are\n"; for(i=0; i&lt;n; i++) { cout&lt;&lt;mat[i][i]&lt;&lt;"\t"; s = s + mat[i][i]; } cout&lt;&lt;"\nSum of major diagonal elements is: "; cout&lt;&lt;s; return 0; }</pre>

### Multi-dimensional arrays

A 2D array where each element is another array is called a **3D(Three Dimensional)** array.

Declaration syntax: data\_type array\_name[size\_1][size\_2][size\_3];

Similarly, more sizes can be specified while declaring arrays to form multidimensional arrays.

## CHAPTER 9

### String Handling and I/O Functions

#### String handling using arrays

A character array can be used to store a string. A null character '\0' is stored at the end of the string. This character is used as the string terminator.

#### Memory allocation for strings

The memory required to store a string will be equal to the number of characters in the string plus one byte for null character.

```
char string1[10]="Bill"; (10 bytes will be allocated, null character is added as  
delimiter for the string, last 5 bytes will be unused)
```

```
char string2[ ]="Program"; (8 bytes will be allocated - 7 for the string and 1 for '\0')
```

#### Input/Output operations on strings

By using input operator >> , we can input only one word.

Eg:

```
char str[20];  
cin>>str;  
cout<<str;
```

If we input Higher Secondary ,the output will be Higher.

**gets()** function is used to input string containing white spaces.

Eg:

```
char str[20];  
gets(str);  
cout<<str;
```

If we input Higher Secondary, the output will be Higher Secondary.

**puts()** function is used to display a string data on the standard output device (monitor).

Eg :

```
char str[10] = "friends";  
puts("hello");  
puts(str);
```

The output of the above code will be as follows:

```
hello  
friends
```

#### 1. Console functions for character I/O

- **getchar()** - used to input a character  
Header file : cstdio  
Syntax : char variable=getchar(); OR getchar(char variable);  
Eg: char ch=getchar(); // the character input through the keyboard is stored in the variable ch
- **putchar()** - used to display a character  
Header file : cstdio  
Syntax : putchar ( variable / character constant )  
Eg: char ch = 'B';  
putchar(ch); // displays B  
putchar('C'); // displays C

#### 2. Stream functions for input / output operations

- **get()** - used to input a single character or stream of characters  
Header file : iostream

Syntax : `cin.get(variable) ; cin.get(array name , size) ;`

Eg: `cin.get(ch) ; // accepts a single character`

`cin.get( str10) ; // accepts a string of maximum 10 characters`

- **getline( )** - used to input a string

Header file : `iostream`

Syntax : `cin.getline(array name , len) ;`

Eg : `cin.getline(str,10); // accepts a string of maximum 10 characters`

`cin.getline(str,10,'c') ; // accepts a string or maximum 10 characters or a string up to the character 'c'`

- **put( )** - used to display a character

Header file : `iostream`

Syntax : `cout.put(variable or character constant) ;`

Eg : `char ch='B' ;`

`cout.put(ch) ; // displays B`

`cout.put('P') ; // displays P`

- **write( )** - used to display a string

Header file : `iostream`

Syntax : `cout.write(arrayname , len) ;`

Eg : `char str[20]="HELLO FRIENDS" ;`

`cout.write(str,5) ; // displays HELLO`

ACT MALAPPUURAM

## CHAPTER 10 FUNCTIONS

### Modular Programming / Modularization

The process of breaking large programs into smaller sub programs is called modularization.

### Merits of Modular programming

- Reduce the size of the program
- Reduce program complexity
- Less chance of error
- Improve re usability

### Demerits of Modular programming

- Proper breaking down of the problem is a challenging task
- Each sub program should be independent

### Function

Function is a named unit of statements in a program to perform a specific task.

main() is an essential function in C++ program . The execution of the program begins in main().

Two types of functions

- Predefined functions / built-in functions - ready to use programs
- User defined functions

### Arguments / Parameters

The data required to perform the task assigned to a function. They are provided within the pair of parentheses of the function name.

### Return value

The result obtained after performing the task assigned to a function. Some functions do not return any value

### Built-in functions

<i>Type &amp; Header File</i>	<i>Function</i>	<i>Use</i>	<i>Syntax</i>	<i>Example</i>
String (cstring)	<b>strlen( )</b>	find the length of the string ( number of characters)	int strlen(string);	strlen("COMPUTER"); // displays 8
	<b>strcpy( )</b>	copy a string to another	strcpy(string1, string2);	char str[20]; strcpy( str,"welcome"); //the string "welcome" will be stored in the variable str
	<b>strcat( )</b>	append one string to another (join two strings)	strcat(string1,string2);	cout<<strcat("Welcome " , "to C++"); // displays Welcome to C++
	<b>strcmp( )</b>	compare two strings	strcmp(string1,string2);	cout<<strcmp("Hello","Hello"); // displays 0
	<b>strcmpi()</b>	compare two strings ignoring the cases	strcmpi(string1,string 2)	cout<<strcmpi("HELLO" , "Hello"); // displays 0
Mathematical (cmath)	<b>abs( )</b>	find the absolute value of a number	int abs( int) ;	cout<<abs(-5) ; // displays 5
	<b>sqrt( )</b>	Find square root of a number	double sqrt(double) ;	cout<<sqrt(16) ; // displays 4
	<b>pow( )</b>	find the power of a	double pow(double,	cout<<pow(3,2) ;

		number	double);	// displays 9
	<b>sin()</b>	find the sine value of an angle.	double sin(double); (angle must be in radians)	cout<<sin(90*3.14/180); displays 1
	<b>cos()</b>	Find the cosine value of an angle	double cos(double); (angle must be in radians)	cout<<cos(0*3.14/180); displays 1
Character (ctype)	<b>isupper( )</b>	to check whether a character is in upper case(capital letter) or not.	int isupper(char);	cout<<isupper('A'); // displays 1 cout<<isupper('d');
	<b>islower( )</b>	check whether a character is in lower case(small letter) or not.	int islower(char);	cout<<islower('A'); // displays 0 cout<<islower('d');
	<b>isalpha( )</b>	check whether a character is an alphabet or not.	int isalpha(char);	cout<<isalpha('B'); // displays 1 cout<<isalpha('8');
	<b>isdigit( )</b>	check whether a character is digit or not	int isdigit(char c);	cout<<isdigit('8'); // displays 1 cout<<isdigit('b');
	<b>isalnum()</b>	check whether a character is alphanumeric or not.	int isalnum(char);	cout<<isalnum('8'); // displays 1 cout<<isalnum('a'); // displays 1 cout<<isalnum('+');
	<b>toupper()</b>	convert the given character into its uppercase.	char toupper(char);	cout<<(char)toupper('b');
	<b>tolower( )</b> -	convert the given character into its lower case.	char tolower(char);	cout<<(char)tolower('B');
Conversion functions cstdlib	<b>itoa()</b>	convert an integer value to string type.	itoa(int n, char c[], int len);	int n = 2345; char c[10]; itoa(n, c, 10); cout << c; // displays "2345"
	<b>atoi()</b>	returns the integer value of the string	int atoi(char c[]);	int n; char c[10] = "2345"; n = atoi(c); cout << n; // displays number 2345
I/O Manipulating function (ciomanip)	<b>setw()</b>	set the width for the subsequent string	setw(int);	char s[]="hello"; cout<<s<<setw(10)<<"friends"; // hello friends

## User defined functions

The syntax of a function definition is given below:

```
data_type function_name(argument list)
{
    statements in the body;
}
```

The data\_type is any valid data type of C++. The function\_name is a user defined word (identifier).

The argument list is a list of parameters, i.e. a list of variables preceded by data types and separated by commas.

Example 1	Example 2	Example 3	Example 4
<pre>int sum(int a , int b) {     int s=a+b ;     return s; }</pre>	<pre>void sum(int a , int b) {     int s=a+b ;     cout&lt;&lt;"Sum="&lt;&lt;s; }</pre>	<pre>void sum(i) {     cin&gt;&gt;a&gt;&gt;b;     int s=a+b ;     cout&lt;&lt;"Sum="&lt;&lt;s; }</pre>	<pre>int sum(i) {     cin&gt;&gt;a&gt;&gt;b;     int s=a+b ;     return s ; }</pre>

## Prototype of functions

A function prototype is the declaration of a function

Syntax : data\_type function\_name(argument list);

Eg : int sum( int , int );

void sum( );

int sum( );

## Arguments of functions

- Arguments or parameters are the means to pass values from the calling function to the called function.
- The variables used in the function definition as arguments are known as **formal arguments**.
- The variables used in the function call are known as **actual (original) arguments**.

## Methods of calling functions

- **Call by value (Pass by value) method** - a copy of the actual argument is passed to the function.
- **Call by reference (Pass by reference) method** - the reference of the actual argument is passed to the function.

## Difference between Call by value method and Call by reference method

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

### **Scope and life of variables and functions**

Local variable - declared within a function or a block of statements.

Global variable - declared outside all the functions.

Local function - declared within a function or a block of statements and defined after the calling function.

Global function - declared or defined outside all other functions.

ACT MALAPPURAM

# CHAPTER 11

## Computer Networks

- It is a group of computers and other devices connected to each other electronically through a communication medium

### Need for Network

#### Resource sharing

- Any hardware /software resource in one system can be shared with other systems in the network.

#### Price performance ratio

- The cost of purchasing licensed software for each computer, can be reduced by purchasing network versions of such software in a network.

#### Communication

- The computer network helps users to communicate with any other computer in the network through its services like e-mail, chatting, video conferencing, etc.

#### Reliability

- In a network, it is possible to backup data on multiple computers. This helps users to retrieve data in the case of failures in accessing data.

#### Scalability

- Computing capacity can be increased or decreased easily by adding or removing computers to the network.

### Some key terms

**Bandwidth:** It measures the amount of data that can be sent over a specific connection in a given amount of time.

**Noise:** It is unwanted electric or electromagnetic energy that lowers the quantity of data signals.

**Node:** Any device which is directly connected to a network is called a Node.

### Data Communication System

- Data Communication is the exchange of digital data between any two devices through a medium of transmission.
- 5 basic elements:

**Message:** It is the information to be communicated.

**Sender:** The computer or devices that is used to sending messages.

**Receiver:** The computer or device that receives the messages.

**Medium:** It is the physical path through which a message travels from the sender to the receiver.

**Protocol:** The rules under which message transmission takes place between the sender and the receiver.

### Communication Medium

- The medium for data transmission over a computer network is called communication channel or communication medium.
- Two types: (1) Guided (2) Unguided

### Guided Medium

- Coaxial cable, twisted pair cable (Ethernet cable) and optical fibre cable

#### (a) Twisted pair cable (Ethernet cable)

- Most widely used
- Two types: (1) Unshielded Twisted pair (UTP), (2) Shielded Twisted Pair (STP)

#### *Characteristics of UTP cable*

- Low cost
- Thin and flexible
- Easy installation



- Carry data upto a length of 100m

#### ***Characteristics of STP cable***

- Better immunity against noise
- Expensive
- Difficult to install
- RJ-45 connector is used to connect UTP/STP twisted pair cable to a computer.

#### **(b) Coaxial cable**

- Carries data to long distance 185-500m
- High band width
- Less noise
- Thicker than twisted pair
- Less flexible
- Difficult to install

#### **(c) Optical fibre cable**

- Data travels as fast as light
- High bandwidth
- Carries data for a long distance
- It uses light for data transmission
- Most expensive and efficient
- Installation and maintenance are difficult

### **Unguided Medium**

- Electro magnetic waves are used for wireless communication

#### **(a) Radio Waves**

- Frequency range is 3KHz to 3 GHz
- Used for short and long distance communication
- Waves are transmitted in all directions
- Inexpensive than wired media
- Can penetrate through most objects
- Transmission can be affected by motors and other electric equipments
- Less secure
- Permission required for radio wave transmission

#### **(b) Microwaves**

- Frequency range is 300MHz to 300GHz
- Travels in straight line
- Can not penetrate any solid object
- Inexpensive than wired media

#### **(c) Infrared Waves**

- Frequency range is 300GHz to 400THz
- Used for short range communication
- Line of sight transmission
- Only two devices can communicate at a time
- Can not cross solid objects
- The longer the distance the weaker the performance

### **Wireless Communication technologies using radio waves**

#### **(a) Bluetooth**

- Frequency range is 2.402GHz to 2.480GHz
- Used for short distance communication
- Used in cell phones, laptops, mouse, keyboard etc.
- It cannot connect upto 8 devices simultaneously
- Slow data transfer rate

**(b) Wi-Fi**

- Frequency range is 2.4GHz to 5GHz
- Data transmission speed upto 54Mbps
- It can be used to connect more number of devices simultaneously
- Used for communication upto 375ft

**(c) Wi-MAX**

- Worldwide Interoperability for Microwave Access
- Frequency range is 2GHz to 11GHz
- Used to provide high speed wireless internet access over very long distances
- Hundreds of users can connect to a single station
- High speed connection upto 70Mbps
- Weather conditions may interrupt the signal
- High power consumption

**(d) Satellite link**

- Uses Geostationary satellites
- These satellites contains an electronic device called transponder
- Transponders are used to amplify and re-broadcast signals to earth
- Transmission of signals from earth to a satellite is called uplink
- Transmission of signals from satellite to earth is called Downlink
- Uplink frequency is 106GHz to 30.0GHz
- Downlink frequency is 1.5GHz to 20.0GHz
- It can cover a large area of the earth
- Expensive
- Requires legal permission and authorisation

**Data Communication Devices**

- Its an interface between computer and the communication channels

**Network Interface Card(NIC)**

- It breaks up data into manageable units
- Translate the protocol
- Some NIC cards have wireless connection
- Data transfer rate is 1Gbps

**Hub**

- Used to connect devices of same network
- It transmit the packet to all other computers connected to it
- It increases the network traffic and reduces the effective band width

**Switch**

- A switch is a device that connects several computers to form a network.
- It an intelligent device, because it can transmit the received data to the destination only.
- It will store the addresses of all the devices connected to it .

**Repeater**

- Used to regenerates the signals through a communication medium

**Bridge**

- A bridge is a device used to segmentize a network.
- A network can be split into different segments and can be interconnected using a bridge.
- This reduces the amount of traffic on a network.

**Router**

- A router is a device that can interconnect two networks of the same type using the same protocol.
- It can find the optimal path for data packets to travel and reduce the amount of traffic on a network.

### Gateway

- A gateway is a device that can interconnect two different networks having different protocols.
- It can translate one protocol to another.
- It can find the optimal path for packets to reach the destination.

### Data Terminal equipments

- Data terminal equipment is a device that controls data flowing to or from a computer.

### Modem

- A modem is a device used for communication between computers through telephone lines.
- The name is formed from **modulator and demodulator**.
- It converts digital signals received from a computer to analog signals for telephone lines and vice versa.

### Multiplexer/Demultiplexer

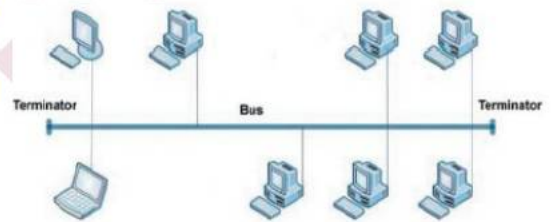
- Multiplexing divides the physical medium into logical segments called frequency channels
- Multiplexer combines the inputs from different sources and sends them through different channels of a medium
- A demultiplexer separates the signal and send them to destination

### Network Topologies

- The way in which the nodes are physically interconnected to form a network.
- Major topologies are bus, star, ring and mesh

### Bus Topology

- In bus topology, all the nodes are connected to a main cable called bus.
- A small device called a terminator is attached to each end of the bus.
- If a node has to send data to another node, it sends data to the bus. The signal travels through the bus and each node checks the bus and only the intended node will accept the data.

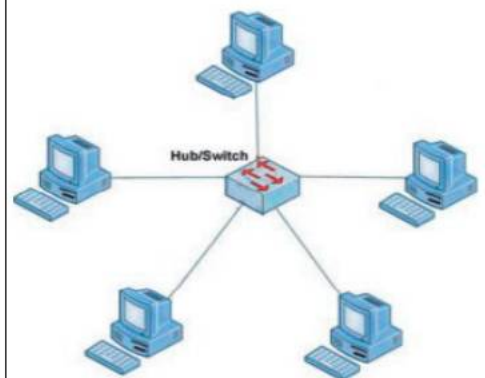


### Characteristics of bus topology

- 1) Easy to install.
- 2) Requires less cable length and hence it is cost-effective.
- 3) Failure of a node does not affect the network.
- 4) Failure of cable (bus) or terminator leads to a breakdown of the entire network.
- 5) Fault diagnosis is difficult.
- 6) Only one node can transmit data at a time.

### Star Topology

- In star topology, each node is directly connected to a hub/switch.
- If any node has to send some information to any other node, it sends the signal to the hub/switch.
- The signal is then broadcasted (in the case of a hub) to all the nodes but is accepted only by the intended node.
- In the case of a switch, the signal is sent only to the intended node..

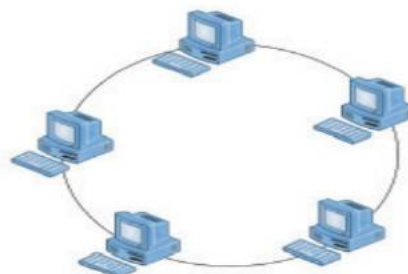


### Characteristics of star topology

- 1) More efficient compared to bus topology.
- 2) Easy to install.
- 3) Easy to diagnose faults.
- 4) Easy to expand depending on the specifications of the central hub/switch.
- 5) Failure of hub/switch leads to failure of the entire network.
- 6) Requires more cable length compared to bus topology.

### Ring Topology

- In ring topology, all nodes are connected using a cable that loops the ring or circle.
- A ring topology is in the form of a circle.
- Data travels only in one direction in a ring.
- Each node regenerates the signal and passes to the next node until it reaches the intended node reaches.

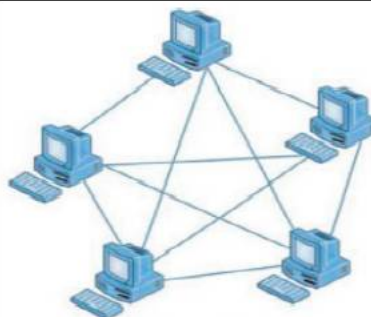


### Characteristics of ring topology

- 1) No signal amplification is required as each node amplifies the signal.
- 2) Requires less cable length and hence is cost-effective.
- 3) If one node fails, the entire network will fail.
- 4) Addition of nodes to the network is difficult.

### Mesh Topology

- In mesh topology, every node is connected to other nodes.
- There will be more than one path between two nodes.
- If one path fails, the data will take another path and reach the destination.



### Characteristics of mesh topology

- 1) Network will not fail even if one path between the nodes fails.
- 2) Expensive because of the extra cables needed.
- 3) Very complex and difficult to manage.

### Types of network

- On the basis of the area covered, computer networks are classified as:  
PAN - Personal Area Network  
LAN - Local Area Network  
MAN - Metropolitan Area Network  
WAN -Wide Area Network

#### Personal Area Network (PAN)

- PAN is a network of communicating devices (computer, mobile, tablet, printer, etc.) in the proximity of an individual.

- It can cover an area of a radius of few meters.

### **LAN (Local Area Network)**

- LAN is a network of computing and communicating devices in a room building or campus.
- It can cover an area of few meters to few kilometers.

### **MAN (Metropolitan Area Network)**

- MAN is a network of computing and communication devices within a city.
- It can cover an area of a few kilometers to a few hundred kilometers radius.
- MAN is usually formed by interconnecting a number of LANs and individual computers.

### **WAN (Wide Area Network)**

- WAN is a network of computing and communicating devices crossing the limits of a city, country, or continent.
- It can cover an area of hundreds of Kilometers in radius.

#### **Summary of PAN, LAN, MAN, WAN**

<b>Parameter</b>	<b>PAN</b>	<b>LAN</b>	<b>MAN</b>	<b>WAN</b>
<b>Area covered</b>	Small area (Up to 10 m radius)	A few meters to few Km (Up to 10 Km radius)	A city and its vicinity (Up to 100 Km radius)	Entire country, continent, or globe
<b>Transmission speed</b>	High speed	High speed	Moderate speed	Low speed
<b>Networking cost</b>	Negligible	Inexpensive	Moderately expensive	Expensive

### **Logical classification of networks**

- Based on the role of the computer in the network
- Two categories: peer to peer and client server

#### **Peer to peer**

- No dedicated servers
- Any computer can act as a client or as a server at any instance
- Ideal for small networks

#### **Client-Server**

- The client server architecture consists of high end computer called server.
- Classification of servers are:

File server: A computer that holds and manage files on a network

Web server: A computer dedicated to responding to requests for webpages

Print server: Redirect the print job from the clients to specific printers

Database server: Allow clients to modify, view or delete data in a common database

### **Network protocols**

- Its a set of rules to be followed in a network when devices in the network exchange data each other

### **TCP/IP**

- Transmission Control Protocol/Internet Protocol
- Used to interconnect network devices on the local networks and then internet
- The data is first broken in to smaller packets by TCP then send
- Internet Protocol(IP) is responsible for the delivery of each of these packets to the right destination
- HTTP, FTP and DNS are the three sub protocols of TCP/IP protocol

## HTTP

- Hypertext Text Transfer protocol
- It is a standard protocol for transferring request from client side and to receive responses from server side
- The pair of HTTP request and response is called HTTP session
- HTTP is transmission medium independent
- Its a stateless protocol

## FTP

- File Transfer Protocol
- It is the easiest way to transfer files between computers over the internet
- A FTP client program can be used to upload and download files(FileZilla, CuteFTP etc.)

## DNS

- Domain Name System
- Returns the IP address of the domain name that we type in the browser's address bar
- DNS maintain a database to store domain names and IP address information of all websites on the internet

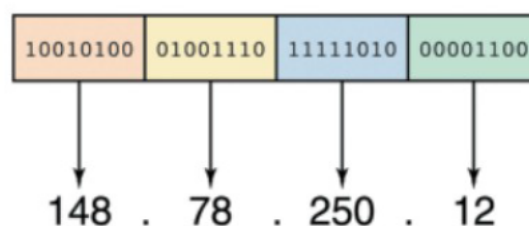
## Identification of computers over a network

### Media Access Control (MAC) addresses

- A universally unique address (12 digit hexadecimal number) assigned to each NIC (Network Interface Card) by its manufacturer.
- MAC addresses are usually written in one of the following two formats:
- MM : MM : MM : SS : SS : SS or MM - MM - MM - SS - SS - SS
- The first half (MM:MM:MM) of a MAC address contains the ID number of the adapter manufacturer.
- The second half (SS:SS:SS) of a MAC address represents the serial number assigned to the adapter (NIC) by its manufacturer.
- eg. 00:A0:C9 : 14:C8:35

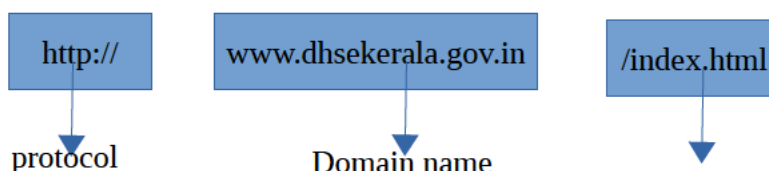
### Internet Protocol (IP) Address

- An IP address is a unique 4 part numeric address assigned to each node on a network, for their unique identification.
- An IP address is a group of four bytes (or 32 bits) each of which can be a number from 0 to 255.



### Uniform Resource Locator(URL)

- Its a formatted string used to identify a network resource on the internet
- URL consists of letters, numbers and punctuations
- Three parts :
  - (1) Network Protocol
  - (2) Domain name
  - (3) Filename



File name

**Protocol**

- Indicate the protocol is used to access the information specified in the domain

**Domain Name**

- It is the name assigned to a server through domain name system

**File Name**

- It is the file to be opened

<b>Generic Domain Names</b>		<b>Country Specific Domain Names</b>	
.com	Commercial Business	.in	India
.edu	Educational Institutions	.au	Australia
.gov	Government agencies	.ca	Canada
.mil	Military	.ch	China
.net	Network organisations	.jp	Japan
.org	Nonprofit organisations	.us	USA

ACT MALAPPURAM

## CHAPTER 12

### Internet and Mobile Computing

#### History of internet

- Internet is developed by United States Department of Defence by the name ARPANET (Advanced Research Project Agency Network)
- ARPANET uses TCP/IP protocol for communication
- ARPANET considered as first wide area connection
- Vincent Gray Cerf is considered as the father of internet
- Tim Berners Lee proposed the idea of World Wide Web(WWW)
- He and his team are credited with inventing HTTP and HTML
- **Internet** is an interconnected system of computer networks that serves the users all over the world
- **Intranet** is a private computer network similar to internet
- When an intranet is made accessible to some computers that are not part of a company's private network it is called an **extranet**

#### Connecting the computer to the internet

- The following hardware and software requirements are required to connect a computer to the internet
  1. A computer with network interface card
  2. Modem
  3. Telephone connection
  4. An internet account given by an Internet service provider(ISP)
  5. Software like Browser

#### Types of connectivity

##### Dial-up Connectivity

- Uses conventional telephone line and a dial up modem to dial and connect to the server at the Internet Service Provider
- These type of connection uses a 56kbps modem
- Maximum speed is 56kbps
- Less costly
- Requires exclusive use of telephone connection

##### Wired broadband Connectivity

- 'always on' connections
- do not need to dial and connected
- Uses a broadband modem and allow us to use the telephone even while using the internet

Dial-up connection	Wired broadband connection
Slow connection, speed upto 56kbps	High speed connection , speed higher than 256kbps
Requires dialling to connect to ISP	Always on connection
Exclusive use of telephone connection	Simultaneous use of voice and internet
Uses dial-up modem	Uses broadband modem

- Popular broadband technologies are Integrated Service Digital Network(ISDN), Cable Internet, Digital Subscriber Line(DSL), Leased Lines and Fiber to the Home(FTTH)

##### 1. Integrated Services Digital Network(ISDN)

- Capable of transporting voice and database



- Uses two lines, one for voice and other for data
- Data transfer rates upto 2Mbps

## 2. Cable Internet

- Uses coaxial cables laid for television signal transmission to our home
- Uses a cable modem
- It can provide speed between 1Mbps to 10Mbps

## 3. Digital Subscriber Line (DSL)

- Uses standard telephone lines
- Use the copper telephone lines for both internet communication and for making the voice calls simultaneously
- ADSL(Asymmetric Digital Subscriber Line) is the common subcategory
- Speed ranges from 256Kbps to 24Mbps
- Most popular broadband service available in india

## 4. Leased line

- It is the dedicated lines used to provide internet facility
- Speed ranges from 2Mbps to 100Mbps

## 5. Fibre To The Home (FTTH)

- Uses optical fibers for data transmission
- High speed connectivity

## Wireless broadband connectivity

- Popular wireless broadband accesses are Mobile Broadband, Wi-Max, Satellite Broadband and Wi-Fi

### 1. Mobile Broadband

- Its a wireless internet access using mobile phone
- The modem is built into the devices like mobile phones, tablet etc.
- Offer the freedom to allow the user to access the internet from anywhere on the move
- It uses the cellular network of the mobile phones for data transmission

### 2. Wi-Max

- Worldwide Interoperability for Microwave Access
- Its an alternative for wired broadband
- It can provide wireless internet access upto a distance of 50KM
- It can provides speed upto 70Mbps

### 3. Satellite broadband

- Internet connectivity is provided through satellite
- A very small aperture Terminal (VSAT) dish antenna and a transceiver (Transmitter and Receiver) are required at the users location
- Speed upto 1Gbps
- Most expensive forms of broadband internet
- Used by banks, stock exchanges, governments etc.

## Internet Access sharing methods

### 1. Using LAN

- The internet connection in a computer can be shared among other computers in the LAN
- It can done using the features of the Operating system or using proxy server software

### 2. Using Wi-Fi network

- The internet connectivity can shared using Wi-Fi router or wireless network access point, popularly called hotspot
- It has a range of about 100 meters
- Less secure than wired connection

### 3. Using Li-Fi network

- Fast optical version of wifi

- Data rate is 100Mbps
- It can be used in aircrafts and hospitlas

## Services on Internet

- The internet offers a variety of services like WWW, e-mail, search engines, social media.

### 1) World Wide Web (WWW)

- WWW is a huge client-server system consisting of millions of clients and servers connected together.

#### a) Browser :

- A web browser is a software that we use to retrieve or present information and to navigate through web pages in the World Wide Web.
- Some common browsers are Google Chrome, Internet Explorer, Mozilla Firefox, Opera, and Safari.

#### b) Web browsing:

- Traversing through the web pages of World Wide Web is called web browsing.

### 2) Search engines

- Internet search engine websites are special programs that are designed to help people to find the information available in World Wide Web.
- Search engine programs search documents available on the World Wide Web for specified keywords.
- It returns a list of the documents/web pages matching the keywords.
- Some of the most popular web search engine sites are Google, Bing, Yahoo Search, Ask, etc.

### 3) E-Mail

- Electronic mail or e-mail is a method of exchanging digital messages between computers over Internet.

#### Sections of an e-mail

To (Recipient Address), Cc (Carbon copy), Bcc (Blind carbon copy), Subject, Content

#### Advantages of using e-mail

Speed, Easy to use, Provision of attachments, Environment friendly, Reply to an e-mail, Cost-effective, Available anywhere anytime

#### Disadvantages of using e-mail

E-mails may carry viruses, Junk mails

### 4) 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.

#### Classification of social media.

##### a) Internet forums

- It is an online discussion website where people can engage in conversations in the form of posted messages.
- e.g. Ubuntu Forum.

##### b) Social blogs

- It is a discussion or informational website consisting of entries or posts displayed.e.g. Blogger and WordPress

##### c) Microblogs

- Microblogs allow users to exchange short sentences, individual images or video links. e.g. Twitter.com

##### d) Wikis

- Wikis allow people to add content or edit existing information in a web page, to form a community document.e.g.wikipedia.org

##### e) Social networks

- Social networking sites allow people to build personal web pages and then connect with friends to communicate and share content. e.g. facebook.com and LinkedIn.

#### **f) Content communities**

- Content communities are websites that organise and share contents like photos, videos, etc. e.g. YouTube

#### **Advantages of social media**

- Bring people together, Plan and organise events, Business promotion, Social skills

#### **Limitations in use of social media**

- Intrusion to privacy, Addiction, Spread rumours

#### **Cyber Security**

- It is used to provide protection of valuable information such as credit card information from unauthorized access.

##### **1) Computer Virus**

- A virus is a bad program to damage routine working of a computer system.
- A computer virus is a program that attaches itself to another program or file enabling it to spread from one computer to another without our knowledge and interferes with the normal operation of a computer.

##### **2) Worm**

- It is a stand alone malware program that replicates itself in order to spread to other computers.
- Worms spread from computer to computer on its own.

##### **3) Trojan Horse**

- It appears as a useful software but it is harmful software's and it will delete useful softwares or files.

##### **4) Spams**

- Sending an email without recipients consent to promote a product or service is called spamming. Such an email is called spam

##### **5) Hacking**

- It is a process of trespassing computer networks.
- In computer networking, hacking is a technical effort to manipulate the normal behavior of network connections and connected systems.
- Hacking performed by computer security experts are called ethical hacking and such hackers are called white hats
- Computer criminals break into secure networks to destroy data or make the network unusable. Such hackers are called black hats
- Grey hat hackers fall between white and black hackers

##### **6) Phishing**

- It is an attempt to get other information such as username, password, bank a/c details etc by posing as the original website, mostly that of banks and other financial institutions.
- Phishing websites have URLs and home pages similar to their original ones.
- The act of creating such a misleading website is called **spoofing**.

##### **7) Denial of Service attack**

- Due to this attack the computer forced to restart and this result refusal of service to the genuine users.

##### **8) Man in the Middle attacks**

- It is an attack in which the an attacker secretly intercepts electronic messages send by the sender to the receiver and then modifies the messages and retransmit it to the receiver.

#### **Preventing network attacks**

##### **1) Firewall**

- A firewall is a system of computer hardware and software that provides security to the computer network in an organisation.

- A firewall controls the incoming and outgoing network traffic by analysing the data

## 2) Antivirus scanner

- It is a tool used to scan computer files for viruses, worms and Trojan horses and cure the infected system.
- When an antivirus program scans a file and notices that the file matches a known piece of malware, the antivirus program stops the file from running, and puts it into 'quarantine'.
- Quarantine is a special area for storing files probably infected with viruses.

## 3) Cookies

- Cookies are small text files that are created when we visit a website that keep track of our details.
- Cookies remembers our user name, preferences, e-mail address, etc.

## Guidelines for using computers over Internet

- Do not open any e-mail attachment that you are not sure about the sender.
- Download files only from reputed sources.
- Do not use/copy software that you cannot confirm the origin.
- Avoid clicking on pop-up advertisements.
- Make sure the firewall is set and turned on.
- Use strong passwords. Change passwords at regular intervals.
- Update the virus definitions of your antivirus program periodically online.
- Keep a regular backup of your important files.
- Be careful about giving personal data online.

## Mobile computing:

- It is a technology that has computing capability and can transmit/receive data while in motion.
- It requires devices like smart phones, laptop etc and wireless internet connection.

### (a) First generation networks ( 1G ):

- They are considered as first analogue wireless telephone technology developed around 1980.
- They are designed for voice calls.

### (b). Second generation networks ( 2G ):

This network introduced Multi Media Service(MMS) and message service. GSM ( Global System for Mobile ), and CDMA ( Code Division Multiple Access )

#### GSM:

- GSM is a globally accepted standard for mobile transmission.
- It uses narrow band TDMA (Time Division Multiple Access )allows simultaneous calls on the same radio frequency rang of 900 MHz to 1800 MHz.
- The network is identified using the SIM( Subscriber Identity Module ).

#### GPRS (General Packet Radio Services )

- It is a packet oriented data service on GSM

#### EDGE (Enhanced Data rates for Global Evolution )

- It is a digital mobile phone technology that is three times faster than GPRS.
- It is used for voice communication and internet connection.

#### CDMA (Code Division Multiple Access ).

- It provides wider coverage than GSM and provides better security to mobile network, better reception even in low signal strength conditions.
- In CDMA several transmitter can send information at the same time over a single channel.

### (c). Third Generation networks (3G) :

- This network provides high data transfer rates and offers high speed wireless broad band services.
- The access part in 3G network uses WCDMA(Wideband CDMA)

(d). **Fourth Generation networks (4G ):**

- Also called LTE (long Term Evolution ) .
- It provides high speed internet facilities, good quality images and videos than TV.
- The access part in 4G network uses OFDMA (Orthogonal Frequency Division Multiple Access.).
- 4G is also referred as 'MAGIC' (Mobile multimedia Any-where, Global mobility solutions over, Integrated wireless and Customised service).

**Mobile communication services:**

(a). **SMS (Short Message Service):**

- It is the transmission of short text messages to and from a mobile phone, fax machine. When a message is sent,
- The message reaches a SMSC ( Short Message Service Centre), that allows 'store and forward' mechanism .
- It uses protocol SS7 ( Signalling System No. 7).
- The first SMS message was 'Merry Christmas ' .

(b). **MMS (Multimedia Message Services )**

It allows sending multimedia content such as text, picture, audio and video files.

(c) **Global Positioning System (GPS):**

- It is a satellite based navigation system that is used to locate a geographical position anywhere on earth.
- It is used for vehicle navigation, aircraft, ship navigation, fishing etc.
- It is maintained by the US Defence Department and is freely accessible to any one with a GPS receiver.
- They are now integrated with mobile phones.

(d) **Smart cards:**

- It is like an ATM card that have a memory chip which stores and transacts data.
- The data is protected and easy to carry. In mobile communication the smart card technology is used in SIM (Subscribers Identity Module ) for GSM phone systems.
- A SIM card is used as an identification proof.

**Mobile Operating Systems:**

It is the OS used in mobile devices like smart phone, tablet etc.

Eg.

- Android from Google
- iOS from Apple
- BlackBerry OS from Blackberry
- Windows Phone from Microsoft.

**Android OS:**

- It is a Linux based OS designed mainly for touch screen mobile devices.
- The interface of Android OS is based on touch inputs like swiping, tapping, pinching and reverse pinching to manipulate on screen objects.
- Android's popularity is due to its open source nature.

.....

ACT MALAPPURAM