COMPUTER APPLICATION

For Fourth Semester Paper – 4.6

(B.A, B.Com, B.B.M & B.Sc Non Computer Science)

As Per Syllabus of Vijayanagara Sri Krishnadevaraya University, Bellary

MOHAMMED SHAFI SARDAR

Faculty of Computer Science,

Shri Gavisiddheshwar Arts, Science & Commerce College,

Gavimath Campus, KOPPAL

A Complete Reference Material for B.A, B.Com, B.B.M and

B.Sc Non-Computer Science of 4th Semester

SYLLABUS

PAPER 4.6 COMPUTER APPLICATION
(B.Com IV / B.A IV / B.Sc IV Non-Computer Science)

SUBJECT CODE	BCOM 407CA2	IA MARKS	20
No. of Lecturer Hrs / Week	04	Exam Hours	03
Total No. of Lecturer Hours	50	Exam Marks	80

Unit I – Introduction to Programming **12 Hours**

Definition, Problem solving using Computer. Overview of C - Introduction, Importance of 'C', Simple 'C' Program, Basic structure of 'C' Program, Programming Style, Executing 'C' Program. Constants, Variable & Data Types **-** C Tokens, Keywords & Identifiers, Constants, Variables, Data Types, Declaration of Variables, Assigning values to Variables, Defining symbolic constants.

Unit II – Operators & Expressions **12 Hours**

Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operations, Increment and Decrement Operators, Conditional Operators, Bitwise Operators, Special Operators, Arithmetic Expression, Evaluation of Expression, Precedence of Operators, Type conversions, Mathematic Functions.

Unit III – Managing Input and Output Statements **10 Hours**

Reading a Character, Writing a Character, Formatted Input, Formatted Output Statements.

Unit IV – Computer Electronics **08 Hours**

Number System, Binary, Octal, Decimal & Hexa Decimal Number System, Binary Arithmetic – Addition, Substraction using 1's & 2's Complement. Digital Logic AND, OR, NOT, NAND, NOR, Universal Gates.

Unit V – Internet Tools **07 Hours**

Concept of Networking – LAN, WAN, MAN, Internet, Intranet & Extranet, Definition of Webpage and HTML, Website and Web Addresses, Web Browsing Tools – Internet Explorer/Netscape Navigator, ISP's in India and their responsibilities, E-mail, Chatting, Video Conference, Search Engines, Downloading and Uploading Files.

Prescribed Books :

1. P.B.Kottur, Computer Concepts & C Programming 17th Edition
2. Rajaraman V, Fundamentals of Computers, 2nd Edition, Prentice Hall India Ltd.,
3. Ramesh Bangia "Learning Internet and E-mail", Khanna Book Publishing
4. Y P Kanetkar, Let Us C
5. E Balagurusamy, "Programming in ANSI C"

CHAPTER – 1 : INTRODUCTION TO PROGRAMMING

Languages are a means of communication. Normally people interact with each other through a language. On the same pattern, communication with computers is carried out through a language. This language is understood both by user and the machine. The Programming language is the medium of communication between the man and the machine. Programming is a set of instructions given to the computer to perform user defined tasks.

PROBLEM SOLVING

Problem solving by the computer involves the following steps.

1) Problem Definition
2) Analysis
3) Algorithm
4) Flowchart
5) Coding
6) Running the program
7) Debugging
8) Testing
9) Documentation

1] PROBLEM DEFINITION

This is the first step in computer problem solving. The problem solver should understand the problem thoroughly in terms of the requirements. That is, what are the input and output operations to be performed ? The programmer should extract from the problem statement, a set of well-defined and precise tasks that can be carried out.

2] ANALYSIS

The given problem must be analyzed before it is solved. This determines the data items, their types and relationship. We should specify the operations (Arithmetic and logic) to be performed on them and checking the suitability in terms of memory and execution time.

3] ALGORITHM

This is a problem solving technique. It can be defined as a step by step procedure to solve a particular problem. It consists of English like statements. Each statement must be precise and well defined to perform a specific operation. When these statements are carried out for a given set of conditions, they will produce the required results. The word algorithm is derived from the famous Arabic author and mathematician, Abu Jafar Mohammad Ibn Musa Al Khowarizmi. The last two terms of the name termed as Algorithm.

CHARACTERISTICS OF ALGORITHM

Each and every algorithm is characterized by the following five important characteristics.

1] Input : It may accept zero or more inputs

2] Outputs : It should produce at least one output (result)

3] Definiteness : Each instruction must be clear, well-defined and precise. There should not be any ambiguity.

4] Finiteness : It should be a sequence of finite instructions. That is, it should end finite time at one or more levels of complexity. It should be effective whenever trace manually for the results.

5] Effectiveness : This means that operations must be simple and are carried out in a finite time at one or more levels of complexity. It should be effective whenever traced manually for the results.

ALGORITHMIC NOTATIONS

While writing algorithms the following notations are considered.

1) **Name of the Algorithm** : It specifies the problem to be solved.
2) **Step number** : Identification tag of an instruction and it is an unsigned positive integer
3) **Explanatory Comment** : It follows the step number and describes the operation. It should be written within a pair of square brackets.
4) **Termination** : It specifies the end of the algorithm. It is generally a Stop statement and the last instruction in the algorithm.

Example 1

Algorithm to compute the area of circle

Algorithm : Area of Circle
Step 1 : Start
Step 2 : Read radius
Step 3 : [Compute the Area]
Area = 3.142 x radius x radius
Step 4 : [Print the Area]
Print "Area of Circle = ", Area
Step 5 : [End of algorithm]
Stop

Example 2

Algorithm to calculate the simple interest.

Algorithm : Simple Interest
Step 1 : Start
Step 2 : [Read the value P, T, R]
Read P, T, R
Step 3 : [Compute the Simple Interest]
SI = (PxTxR) / 100
Step 4 : [Print the Simple Interest]
Print "Simple Interest= ", SI
Step 5 : [End of algorithm]
Stop

Example 3

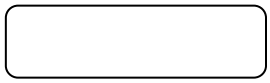
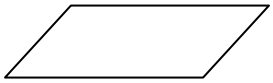
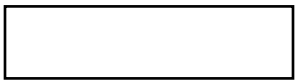
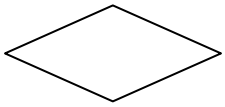
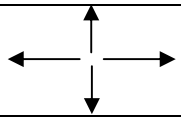
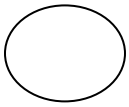
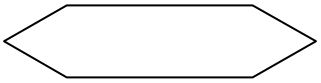
Algorithm to find the largest of two numbers.

Algorithm : Largest of two numbers
Step 1 : Start
Step 2 : [Read the values A and B]
Read A, B
Step 3 : [Compare A and B]
If (A>B) Then
Print 'A is largest'
Else
Print 'B is largest'
End If
Step 6 : [End of algorithm]
Stop

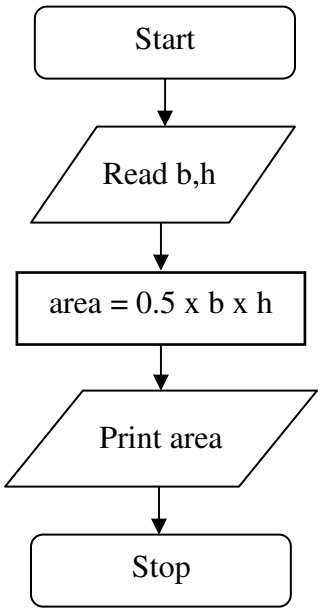
4] FLOWCHART

This is a chart showing a flow of logic involved in solving a problem. This is defined for an algorithm. The flowchart can be defined as a diagrammatic representation of an algorithm. It is referred to as the blue print of an algorithm. It is also defined as a visual or graphical representation of an algorithm. The flowchart is an easy way to understand and analyze the problem. It is a useful aid for programmers and system analysts.

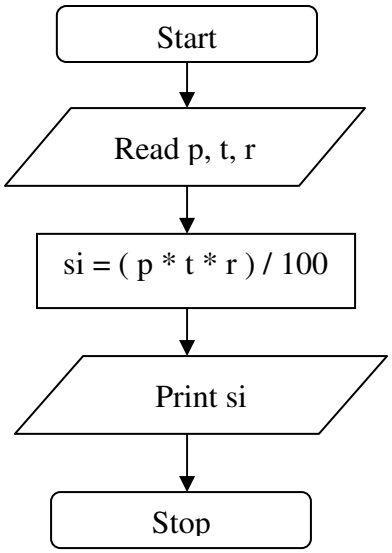
Flowcharts make use of geometric figures, to specify a particular operation. Those are

Geometrical Figure	Name	Function
	Oval	Start and Stop
	Parallelogram	Input and Output
	Rectangle	Processing
	Rhombus	Decision Making
	Arrows	Connections
	Circle	Continuation
	Hexagon	Repetition / Looping

Example 1 : Draw a flowchart to find area of triangle



Example 2 : Draw a flowchart to find the Simple Interest.



5] CODING

The complete structure of a problem to be solved by a computer is called a program. The computer does not process an algorithm or a flowchart, but executes the program. A program is a set of instructions to solve a particular problem by the computer and the actual process of writing a program is called coding. Program are written using programming languages and are fed to the computer.

6] RUNNING THE PROGRAM :

The program can be run(executed) in the central processing unit. This phase of problem solving by a computer involves three steps.

1) Understand the instructions
2) Store data and instructions
3) Perform computations

The user prepares his/her program and makes it ready for execution. All the instructions stored in the RAM, must be fetched one by one to the ALU to perform corresponding operations. This is called the fetch/execute cycle. The processed data is stored again in the RAM. Finally, they are transferred to the output devices.

7] DEBUGGING

The process of detecting and correcting errors(mistakes) in the program is known as debugging. There is a program called debugger that takes object program as input and executes it and helps in eliminating the mistakes that occur in the source program.

Generally, programmers commit three types of errors. These are,

1) Syntax Errors
2) Logical Errors
3) Run-time Errors

SYNTAX ERRORS

This type of errors is the result of violation of programming rules. On encountering these errors a computer displays error message specifying the line number. It is easy to debug these errors. For Example: If the statement of C is typed without the semicolon at the end, then there will be an error because of a missing semicolon.

LOGICAL ERRORS

Logical errors occur during coding process. When the programmer codes his problem, he must take care of correct operations to be performed. The program will be executed but produce some unwanted results. It is very difficult to debug such errors, because the computer does not display them. We can eliminate such errors by tracing it and running for sample data.

RUN-TIME ERRORS

These errors occur when we attempt to run an ambiguous instructions. For example, an infinite loop in program sequence which causes no output. These are also occur due to device errors, improper sequencing of constructs, errors in system software, incorrect data input etc., The computer will print error message. Some of run-time errors are:

1) Divide by zero
2) Null pointer assignment
3) Data overflow

8] TESTING

The process of executing the program to test the correctness of the outputs of the problem is called testing. The program is tested by executing with different sets of data. Logical errors are the outcome of this process.

9] DOCUMENTATIONS

While writing programs, it is good programming practice to make a brief explanatory note on the program or program segments. This explanatory note is called a comment. It explains how the program works and how to interact with it. Thus, it helps other programmers to understand the program.

There are two types of documentation. They are,

1) Internal documentation
2) External documentation

INTERNAL DOCUMENTATION

This documentation is a comment statement within a program. It describes the function of the program or program segments. These statements are not translated to machine language. Translators simply discard these statements during the translation process.

EXTERNAL DOCUMENTATION

This documentation is an executable statement in a program. It may be a message to the user to respond to the program requirement. This is accomplished using output statements. It makes the program more attractive and interactive. Some sample examples are given below:

Print, “Input numbers one by one”

Print, “Input the order of the matrix”

Print, “Do you what to continue ?”

INTRODUCTION TO C

C is a high level language. It is both general purpose and specific purpose programming language. Now a days, C has become a common programming language for every application developer. It was developed at Bell & T Laboratory, USA in 1972. It is the outcome of the efforts of Dennis Ritchie and Brian Kernighan.

C is derived from two early programming languages such as BCPL (Basic Combined Programming Language) and B Language. The BCPL was developed by Martin Richards. And, B was developed by Ken Thompson.

In 1972, Dennis Ritchie developed a new version of B and named it as C. He selected the name C for his new language because C comes after B in the alphabetical order which indicates advancement to B. Some sources also say that Dennis Ritchie selected the second character of BCPL for naming his language.

In 1978, Dennis Ritchie and Brian Kernighan jointly published a detailed description of the C Language document. It is known as "K & R C". It was released to commercial applications in 1978.

CHARACTERISTICS OF C

C has become popular programming language because of its many features. The important characteristics of C are :

- C is a general purpose programming language
- C is a structured programming language
- Helps in development of System Software
- It has rich set of operators
- It provides compact representation for expressions
- It allows manipulation of internal processor registers
- No rigid format. Any number of statements can be typed in a single line.
- Portability : any C program can be run on different machines with little or not modification.
- Supports a rich set of data types
- Very less number of reserved words
- Pointer arithmetic and pointer manipulation
- Ability to extend itself by adding functions to its library

APPLICATIONS OF C

Because of its portability and efficiency, C is used to develop the system as well as application software. Some of the system and application software are listed below.

System Software

1] Operating Systems	2] Interpreters	3] Compilers	4] Assemblers
5] Editors	6] Loaders	7] Linkers	

Application Software

1] Data Base Systems	2] Graphic Packages	3] Spread Sheets
4] CAD/CAM Application	5] Word Processors	6] Office Automation Tools
7] Scientific and Engineering Applications		

BASIC STRUCTURE OF A C PROGRAM :

Every programming languages have their own format of coding. The complete structure of C program as shown below.

```
preprocessor statements
global declarations;
main( )
{
        declaration;
        statements;
        /* comments */
}
user defined function
```

1] Preprocessor Statements :

These statements begin with # symbol and also called preprocessor directives. These statements direct the C preprocessor to include header files and also symbolic constants into a C program. Some of the preprocessor statements are :

include <stdio.h>
include <conion.h> etc.,

2] Global Declaration :

Variables or functions whose existence is known in the main( ) function and other user defined functions, are called the global variables and their declarations are called the global declarations. This declaration should be made before main( ) function.

3] main( ) function :

This is the main function of every C program. Execution of C program starts from main(). No C program is executed without main() function. It should be written in lowercase letters and should not be terminated by a semicolon. It calls other library functions and user defined functions.

4] Braces :

Every C program uses a p[air of curly braces i.e., { and }. The left brace indicates the beginning of main( ) function. On the other hand, the right indicates the end of the main( ) function. The braces can also be used to indicate the beginning and end of user-defined functions and compound statements.

5] Declaration :

It is a part of the C program where all the variables, arrays, functions etc., used in the C program are declared and may be initialized with their basic data types.

6] Statements :

These are instructions to the computer to perform specific operations. They may be input-output statements, arithmetic statements, control statements and other statements.

7] Comments :

Comments are explanatory note on some instructions. The statements to be commented on must be enclosed with /* and */. Comment statements are not compiled and executed.

8] User-defined functions :

These are subprograms. Generally, a subprogram is a function. The user defined functions contain a set of statements to perform a specific task. These are written by the user, hence the name user-defined functions. They may be written before or after the main( ) function.

A Simple C Program is given below

```
#include<stdio.h>
main()
{
printf("\nWelcome to C");
\* a call to output function *\
}
```

The first line tells the compiler to include standard input/output header file to perform reading and printing of the data. The second line is the main( ), the main function of a C program. The body of C program contains the statement i.e.,

printf("\nWelcome to C");

When this statement is taken for execution, main( ) calls the output function printf( ). Then printf( ) prints **Welcome to C** on the computer screen."\n" is a new line character. It is used to print the message to the new line. The statement i.e., * a call to output function *\ is comment. Comment describes the above statement.

Program Compilation and Execution

Compiling a C program means translating it into machine language. 'C' compilers are used for this purpose. The process of generating outputs of the program on the screen is called Execution. Press **Ctrl + F9** to compile the program. Press **Alt + F5** to run the program.

Starting C Program Editor

1] Click **Start** menu & Select **Run** option
2] Type **command** & Press **Ente**r key
3] Press **Alt+Enter** key to view DOS in Full Screen
4] Type **cd\tc\bin** & Press **Enter** Key
5] Type **tc** & Press **Enter** Key

CHARACTER SET

Every programming language has its own set of characters to form the lexical elements. The characters used in C are grouped into the following three categories.

1] Alphabets : Upper Case A-Z
Lower Case a-z
2] Digits : 0 to 9
3] Special Characters :

, comma	+ plus sign
. dot	- minu sign
: colon	* asterisk
; semicolon	/ slash
‘ opostrophe	% percentage
“ quotation mark	& ampersand
? question mark	^ caret
! exclamation mark	~ tilde
_ underscore	< less than
# hash	> greater than
= equal sign	\ back slash
| pipeline character	(left parenthesis
	) right parenthesis
	[left bracket
	] right bracket
	{ left brace
	} right brace

C TOKENS

The basic and the smallest units of a C program are called C Tokens. There are six types of tokens in C.

1] Keywords
2] Identifiers
3] Constants
4] String
5] Operators
6] Special Symbols

KEYWORDS

Every word in a C program is either a keyword or an identifier. All keywords (reserve words) are basically the sequence of characters that have one or more fixed meaning. All C keywords must be written in lowercase letters. Because, in C both uppercase and lowercase letters are significant.
Example for Keywords :

int	char	break	else	if	continue
do	float	goto	long	return	for
short	static	auto	union	struct	switch

IDENTIFIERS

Identifiers are names given to the program elements such as variables, arrays and functions. Basically, identifiers are the sequences of alphabets and digits.

Rules for forming identifier name

- The first character must be an alphabet (upper case or lowercase) or an underscore (_)
- All succeeding characters must be either letters or digits
- Uppercase and lower case identifiers are different in C
- No special character or punctuation symbols are allowed except the underscore (_)
- No two successive underscores are allowed
- Keywords should not be used as identifiers.

CONSTANTS :

A Constant is an entity whose value does not change during program execution. The following are types of constants :

1] Integer Constant :

An integer constant is a whole number. It is a sequence of digits without a decimal point. It may prefixed with plus or minus sign.
Example : 246, 0, -3579, +25, -32028, 9999 etc.,

2] Floating Constant :

A floating constant is a number with a decimal point. It is defined as sequence of digits preceded and followed by the decimal point. They may have prefixed plus or minus sign.
Example : -246.01, +12.25, 0.0, 0.0005, 82.0, 9999.999, -0.00123 etc.,

3] Character Constant :

A character constant is a single character enclosed within pair of apostrophes.
Example : ‘a’, ‘?’, ‘#’, ‘ ‘(blank character) etc.,

4] String Constant :

A string constant is a sequence of characters enclosed within a pair of double quotes.
Example : “Hi”, “Welcome”, “2007”, “x+2”etc.,

VARIABLES

A variable is an entity used by a program to store values in the program. It is a symbolic name used for actual memory location. Therefore variable are also called as identifiers.
Example : sum, area, num, length, age, city etc.,

Rules for forming variable names

- The first character of a variable name must be an alphabet or an underscore
- All succeeding characters consists of letters and digits
- Both uppercase and lowercase variables are significant in C
- Keywords should be used as variables
- Special characters are not allowed
- There is not limit on the number of characters in a variable name
- Always choose an appropriate variable name that makes proper sense to the user

DECLARATION OF VARIABLES

All the variables must be declared before they are used in the program. A variable declaring consists of a data type name followed by a list of one or more variables of that type, separated by commas.

Syntax :

Datatype Variable_name,…..,….;

Example :

```
int num;
char name[10];
float rate;
```

ASSIGNING VALUES TO VARIABLES

We know that the variables represent some memory location, where the data is stored. Each variable is associated with one or more values. The process of giving values to variables is called assignment of values. The assignment operator '=' is used to assign a value to a variable.

Syntax :

```
variable_name = value;
```

Example :

```
int num=5;
float pi=3.142;
x = 10;
sum = a + b;
name = "Saraswati"
```

DATA TYPES

Data type indicate the type of data that a variable can hold. The data may be numeric or non-numeric in nature. There are four fundamental C data types

Types	Keyword	Size(in bytes)
Integers	int	2 bytes
Real(Floating point)	float	4 bytes
Double Precision	double	8 bytes
Character	char	1 byte

int :

This is keyword used to indicate an integer number. Any integer number is a sequence of digits without a decimal point. A 8 bit computer the maximum integer that can be input is either -128 or +127. Similarly, a 16 bit computer handles the integers from -32,768 to +32,767.

Example :

-248, 14042, 27246, +1996, 0, 32760

float :

This is a keyword used to indicate a floating-point number. The floating-point numbers are same as real numbers. They are called floating point. These numbers may be expressed in scientific notation.

Example : -263.238, 2.63238E+02, 0.0263238E+04, 26323.802

char :

This is a keyword used to indicate the character type data. The data may be a character constant or a string constant. A character constant can be defined as any single character enclosed within a pair of apostrophes.

Example :

'a', 'p', '$', '2', '?', ' '(blank) etc.,

double :

This is a keyword used to indicate a double precision floating point numbers. The precision is associated with the accuracy of data. The float usually stores a maximum or 6 digits after the decimal point. But double stored 16 significant digits after the decimal point.

Example : 234.0000000000000000, -0.0000001023999001

BACKSLASH CONSTANTS

A backslash constant a combination of two character in which the first character is always the backslash(\ ). And the second character can be any one of the characters a, b, f, n, r, t, v , ', ", \, and 0.The backslash characters also called as escape sequences. These statements are used in output statements.

Backslash Constant	Meaning
\a	System Alarm (bell or beep)
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\"	Double quote
\'	Apostrophe
\0	Null character
\\	Back slash character (\)

SYMBOLIC CONSTANTS

A symbolic constant is a name that substitutes a numeric constant, a character constant or a string constant. For example, assume that we are computing the area of a circle using the formula.

area = 3.142 * radius * radius

In this formula, the numeric constant 3.142 can be replaced by symbolic constant name PI. Thus, the formula takes a new form as

area = PI * radius * radius

Here, we substituted PI for a numeric constant. During compile-time, each occurrence of PI is replaced by its value defined. Symbolic constants are defined using preprocessor statement #define.

Example :
```
# define   PI   3.142
# define   NAME   "PRABHU"
# define   FOUND   1
```

Symbolic constants must be defined at the beginning of a program. That is, before the main( ) function of a C Program.

****

CHAPTER – 2 : OPERATORS AND EXPRESSIONS

OPERATORS IN C

C has a rich set of operators. They may operate on a single operand or two operands. They are used to perform basic arithmetic operations, comparisons, manipulation of bits and so on. C operators are broadly classified into three main categories.

1] Unary Operators
2] Binary Operators
3] Ternary Operators

1] UNARY OPERATORS :

An operator that acts upon only one operand is known as a unary operator. The C unary operators are :

1) Unary minus (-)
2) Logical NOT operator (!)
3) Bitwise Complementation (~)

2] BINARY OPERATORS :

These operators act upon two operands. Hence the name binary. The binary operators are further classified into 4 categories.

1) Arithmetic Operators
2) Relational Operators
3) Logical Operators
4) Bitwise Operators

3] TERNARY OPERATORS :

These operators act upon three operands. Hence it is called ternary operator. It is known as conditional operator.

TYPES OF OPERATORS :

1) Arithmetic Operators
2) Relational Operators
3) Logical Operators
4) Assignment Operators
5) Increment and decrement Operators
6) Conditional Operators
7) Bitwise Operators
8) Special Operators

1] ARITHMETIC OPERATORS :

Arithmetic operators are used to perform the basic arithmetic operations such as addition, substraction, multiplication and division. The modulus operator is added to the list of C arithmetic operators. It is used for finding the remainder after an integer division.

Operation	Operator	Precedence	Associatively
Addition	+	2	L to R
Substraction	-	2	L to R
Multiplication	*	1	L to R
Division	/	1	L to R
Modulus	%	1	L to R

Addition Operator (+) : This operator adds the two data values (operands) that appear on either side of it.

Example : sum = num1+num2;

Substraction Operator (-) : The operand that appears to the right of the operator is substracted from the operand that is the left of it.

Example : net = amount – discount;

Multiplication Operator (*) : The two operands that appear on either side of the operator are multiplied.

Example : mul = a * b;

Division Operator (/) : The value to the left of / is divided by that on the right.

Example : div = x / y;

Modulus Operator (%) : This operator evaluates the remainder of an integer division operation.

Example : rem = x % y;

2] RELATIONAL OPERATORS :

These are used to compare two operands. They define the relationship existing between two constants or variables. They result in either a TRUE or FALSE value. The value of TRUE is nonzero (i.e, 1) and that of FALSE is zero.

Operator	Meaning	Precedence	Associativity
<	Lesser than	1	L to R
< =	Less than or equal to	1	L to R
>	Greater than	1	L to R
> =	Greater than or equal to	1	L to R
= =	Equal to	2	L to R
! =	Not equal to	2	L to R

Example :

If a =100, b=80 Then

(a > b) is true

(a < b) is false

Difference between = and = = operator

= = operator	= operator
1. = = is relational operator.	1. = is assignment operator.
2. = = sign does not change the value of an identifier.	2. = operator changes the value of the variable to its left.

3] LOGICAL OPERATORS :

These operators are used to take decisions. These are three such operators. They are AND, OR and NOT. The result of these operators is either TRUE or FALSE. The logical operators are used to connect one or more relational expressions.

Operator	Meaning	Precedence	Associativity
&& (double ampersand)	Logical AND	2	L to R
‖ (double pipeline)	Logical OR	3	L to R
! (exclamation)	Logical NOT	1	L to R

The logical AND operator is used to perform operation on two operands. It is equivalent to a multiplication operation. The result of a logical AND is true, when both the operands are true; otherwise the answer is false.

Operand1	Operand2	Operand1 && Operand2
False	False	False
False	True	False
True	False	False
True	True	True

The logical OR operator is used to perform operation on two operands. It is equivalent to a addition operation. The result of a logical AND is false, only when both the operands are false; otherwise the answer is true.

Operand1	Operand2	Operand1 && Operand2
False	False	False
False	True	True
True	False	True
True	True	True

The logical NOT is used to obtain the logical complement of the operand. The result of a logical NOT is TRUE when the operand is FALSE and vice-versa.

Operand	! Operand
False	True
True	False

Example : If i = 8, x = 5.1, c='a' Then

1. (i > 6) && (c = = 'a') is true or 1
2. (i <= 6) | | (c = = 'b') is false or 0
3. (i >= 6) && (c = = '97') is true
4. (x < 10) | | (i > 10) is true

4] ASSIGNMENT OPERATORS :

The assignment operator = is used to assign the value of a variable, constant or expression to a variable.

Syntax :

variable = variable / constant / expression ;

Example :

```
a = 10;
b = 10;
sum = a + b
```

C also provider abbreviated or short hand assignment operators.

Syntax :

Variable Operator = Expression;

Example :

n = n + 5; can be written using the short hand assignment
n + = 5; where + = is the abbreviated or compound operator.

5] INCREMENT AND DECREMENT OPERATORS :

INCREMENT OPERATOR :

This operator is used to increment the value of an integer quantity by one. This is represented by '++' (double plus) symbol. This symbol can be placed before or after the integer variable, then

++int_var - indicates pre increment i.e., the value of int_var must be incremented before it is used. (increment and use)

int_var++ - indicates post increment i.e., the value of int_var is used first then increment it (use and increment)

Example :
```
int x, y;
x = 10;
y = ++ x;
```

DECREMENT OPERATOR :

This operator is used to reduce the value of an integer quantity by one. This is represented by '--' (double minus) symbol. This symbol can be placed before or after the integer variable, then

--int_var - indicates decrement the value of int_var first and then use it.

int_var-- - indicates use the value of int_var first and then decrement it.

Example :
```
int x, y;
x = 10;
y = -- x;
```

Difference between post fix and pre fix operators :

Pre-increment/decrement	Post increment/decrement
1. The pre increment or decrement operator first increment or decrement the value of its operand by 1 and then assigns the resulting value to the variable on the left hand side. For example : int a, b; a = 15; b = + + a; Here, the value of a is incremented by 1 and then this value 16 is assigned to b.	2. The post increment or decrement operator first assigns the value to the variable on the left hand side and then increment or decrements the operand by 1. For example : int m, n; n = 2; m = n - - ; Here, the value of n is first assigned to m and then decremented by 1, i.e., n becomes 1, but value of m is 2.

6] CONDITIONAL OPERATOR (TERNARY OPERATOR) :

The ternary operator is a conditional operator in C. There is only one such operator in C. It takes three operands. The symbol '?' is used as a ternary operator in C.

Syntax :
```
Variable = expr1 ? expr2 : expr3 ;
```

The general form of ternary expression is :

```
Variable = (condition) ? true : false ;
```

1. The expr1 is evaluated first.
2. If expr1 is true, expr2 is evaluated and its value assigned to variable.
3. If expr1 is false, expr3 is evaluated and its value assigned to the variable.
4. Either expr2 or expr3 will be evaluated, but never both.

Example : m = 5; n = 10;
big = (m > n) ? m : n ;

In the above example m > n is evaluated first giving the result false. Since the expr1 is false the value of n is assigned to big. Therefore the value of big is 10.

7] BITWISE OPERATORS :

All data items are stored in the computer's memory as a sequence of bits (0's and 1's), and some applications need the manipulation of these bits. Manipulation of individual bits is carried out in machine language or assembly language. To perform, the bitwise operations, C provides six operators. These operators work with int and char type data. They cannot be used with floating-point numbers.

Operators	Symbol Name	Meaning
&	Ampersand	Bitwise AND
|	Pipeline	Bitwise OR
^	Caret	Exclusive – OR (XOR)
~	Tilde	1's complement
<<	Double less than	Left shifting of bits
>>	Double greater than	Right shifting of bits

Note : Bitwise operators cannot be applied to float or double. They can be applied to integers only.

Bitwise AND operator : It is represented by the symbol & (ampersand). The result of AND operation is 1 if both operand bits are 1, otherwise, the result is zero.

Example : x=4 & y=5 results in 4
i.e., binary value of x is 0100
binary value of y is 0101
---
x & y, After AND operation 0100 i.e., 4

Bitwise OR operator : It is represented by the symbol | (pipe). It is also known as Inclusive OR. The result of bitwise OR operation is 1 if either or both corresponding bits has a value 1, otherwise it is zero.

Example : x = 4 and y = 9
i.e., binary value of x is 0100
binary value of y is 1001
---
x | y, After OR operation is 1101 i.e., 13

Bitwise exclusive OR operator : It is represented by the symbol ^ (caret). The result of bitwise exclusive OR operation is 1, if one of the corresponding bit is 1, otherwise it is zero.

Example : x = 4 and y = 9
i.e., binary value of x is 0100
binary value of y is 1001
---
x ^ y, After exclusive OR operation is 1101 i.e., 13

Bitwise Complement operator : The complement operator is represented by ~ (tilde) and is also known as one's complement.

Example : x = 4
i.e., x is 0100 Then
~x is 1011

Bitwise shift operator : The shift operators are used to push the bit patterns to the right or left by the bit number of bits given by their right operand.

The Right Shift operator is of the form :

Syntax :

Expression >> n;

Where n is the number of bit positions to be shifted. However, the right most n bits are lost. Further the left most n bits which are vacant will be filled with 0.

Example : num contains the bit pattern
110011110 then

Num>>1 gives 011001111
Num>>3 gives 000110011
Num>>5 gives 000001100

The Left Shift operator is of the form :

Syntax :

Expression << n,

Where n is the number of bit positions to be shifted. The leftmost n bits are lost and the right most n bits are vacant and filled with zeros.

Example : num contains the bit pattern
11010111 then

Num<<1 gives 10101110
Num<<3 gives 10111000
Num<<5 gives 11100000

8] SPECIAL OPERATORS OF C

C contains the following special operators.

Operator	Description
,	Comma operator
sizeof	Size in bytes
&	Address of operand
*	Access value from the address
. (dot)	Dot operator
→	Arrow operator

COMMA OPERATOR :

The comma operator is basically associated with the **for** statement. It is also used to link two or more related expressions together. In such cases, the expressions are evaluated in a left-to-right fashion. And, we can obtain the value of the combined expression from the value of the rightmost expression.

Example: sum = (a = 12, b = 22, a+b);

There are three expressions namely, a = 12, b = 22 and a + b. The value of the rightmost expression, 33 (i.e., a + b = 12 + 22 = 33) is set to the variable sum.

sizeof( ) OPERATOR :

The sizeof( ) operator returns the size (i.e., number of bytes) of an operand. It is a function and therefore has to be written in lowercase letters. It must precede its operand. The operand may be constant, a variable or a data type. It is normally used to determine the size of arrays and structures. The syntax of sizeof( ) operator is as follows.

sizeof(operand);

Example :

x = sizeof(int);
y = sizeof(sum);

Consider the following table :

Variable declaration	Expression	Value
int num	sizeof(num)	2
long num	sizeof(num)	4
float num	sizeof(num)	4
char ch	sizeof(ch)	1
double num	sizeof(num)	8
int a, b;	sizeof(a+2*b)	2

EXPRESSIONS :

It is a combination of operators and operands that computes to a specific value. There are three types of expressions.

1. Arithmetic Expression
2. Relational Expression
3. Logical Expression

1] ARITHMETIC EXPRESSIONS

An expression involving arithmetic operators is called an arithmetic expression. It takes one or more operands and connects them by an arithmetic operator. These operands are either integers or floating point numbers.

EVALUATION OF ARITHMETIC EXPRESSION

Arithmetic expressions are evaluated from left to right. Expressions involving high priority operators are evaluated first.

Rules for evaluation of arithmetic expressions :

1) If the given expression involves parentheses, then the expressions inside the parenthesis must be evaluated first.
2) If a unary minus is present in the expressions, then the term associated with unary minus must be evaluated before the other expressions.

Precedence and associativity of arithmetic operators

Precedence Level	Operators	Associativity
1	( ) expressions inside the brackets	
2	Exponent	
3	++, - -, unary minus (-)	Right to Left
4	*, /, %	Left to Right
5	+, -	Left to Right

Example :

1. 3abc = 3 * a * b * c
2. 2(a+bc) = 2 * (a + b * c)
3. $\frac{x+y}{z}$ = (x + y) / z
4. $2p^2 + 3p + 1$ = 2 * p * p + 3 * p + 1
5. (2x + 1)(3y + z) = (2 * x + 1) * (3 * y + z)
6. $\sqrt{a+b}$ = sqrt (a + b)
7. $\frac{(a+b)}{(a-b)}$ = (a + b) / (a – b)

Evaluate the following expression :

X = 9 – 12 / (3 + 3) * (2 – 1)
X = 9 – 12 / 6 * 1
X = 9 – 2 * 1
X = 9 – 2
X = 7

2] RELATIONAL EXPRESSIONS

A relational expression in a valid combination of variables, constant and relational operators. Generally relational expressions compare the values of two variables or expressions, through relational operators. These operators are used to construct expressions that evaluate to true or false.

Relational operators, their precedence and associativity for evaluation of an expression.

Operator	Meaning	Precedence	Associativity
<	Lesser than	1	Left to Right
< =	Lesser than or Equal to	1	Left to Right
>	Greater than	1	Left to Right
> =	Greater than or Equal to	1	Left to Right
= =	Equal to	2	Left to Right
! =	Not equal to	2	Left to Right

Syntax : Expression1 Relational operator Expression2

Where, Expression1 and Expression2 are arithmetic expressions, which may be simple constants, variable or combination of them.

Example :

4.5 < = 10 - TRUE
4.5 > 10 - FALSE
A+B = = C + D - TRUE if the sum of (A+B) is equal to (C+D)

3] LOGICAL EXPRESSIONS

Expressions involving the logical operators are called Logical Expressions. The result of a logical expression is either TRUE (Non-Zero) or FALSE (Zero). The logical expression involves AND (&&), OR (||) and NOT (!) operators.

Operator	Meaning	Precedence	Associativity
&& (double ampersand)	Logical AND	2	L to R
|| (double pipeline)	Logical OR	3	L to R
! (exclamation)	Logical NOT	1	L to R

Example : (marks > = 35) && (marks < = 100)
(ch > = ‘A’) & & (ch < = ‘Z’)

TYPE CONVERSION

In some applications, we may often want to change the data type of the variable. When we declare some variable as int, the desired output may be float or vice versa. In such situations, we may change the nature of data stored in a variable. This process is known as data type conversion. It is also called type casting.

The process of converting one predefined type into another is called type conversion. There are two types.

1. Implicit type conversion
2. Explicit type conversion

IMPLICIT TYPE CONVERSION :

This conversion performed by compiler without programmers intervention. This conversion is applied generally whenever different data types are intermixed in an expression.

Assignment	Conversion	Result

int = float	Converts float into int by truncating the fraction part	int x;
		foat y=25.5543;
		x=y;
		value of x is 25.

float = int	Converts int to float by adding zeros to the fractional part	float y; y = 10; value of y is 10.000000

double = float	Converts float to double by addition fractional part with across.	float x = 5.123456;
		double y; y = x;
		The value of y ix 5.1234560000

float = double	Double value is rounded to the precision of float before truncating	float x;
		double y = 2.5678987650;
		x=y;
		The value x is 2.567899

int = char char = int	Assign the ASCII code of char to int. Assign the ASCII code of int to char	int x; x = 'A' value of x is 65 char a; a = 65; value of a is 'A'

EXPLICIT TYPE CONVERSION :

The implicit type conversions are automatic. However, a user can convert data explicitly by type casting.

The general form of type casting is as follows :

(data type) expression;

Where,

data type – any basic data type and must be written within the parentheses.

Example :

n1 = (int) 25.5;
n2 = (int) 20.5 / (int) 5.5;
n3 = (int) (p+q)
n4 = (int) p + q

MATHEMATICAL FUNCTIONS

There are certain functions which are readily available in the language. C itself are called library functions.

C provides a large number of library functions which perform specific operations. In C all mathematical functions are enclosed in the header file math.h. Mathematical functions are also included among these library functions.

Function	Purpose	Usage
sin(x)	Sin of x, x is radians	If d=60, x=d*3.142/180, y=sin(x)
cos(x)	Cosine of x, x in radians	If d=60, x=d*3.142/180, y=cos(x)
tan(x)	Tangent of x, x in radians	If d=60, x=d*3.142/180, y=tan(x)
exp(x)	E raised to the power of x (e = 2.718282)	If x = 2.5, y=exp(x)
log(x)	Natural logarithm of x (base e)	y=log(3.0)
log10(x)	Base 10 logarithm of x	y=log10(3.0)
pow(x, y)	X raised to the power y, i.e., X^Y	y=pow(3,4)
sqrt(x)	Square root of x	y=sqrt(25)
fabs(x)	Absolute value of x	y=fabs(-3.14)
ciel(x)	Rounds x up to the nearest integer	y=ciel(2.339)
floor(x)	Rounds x down to the nearest integer	y=floor(3.339)

Transfer the following mathematical expressions in to C expressions :

1. $\sqrt{a+b+c}$ = sqrt (a + b + c)
2. $(a + b)^4$ = pow (a+b, 4)
3. $Log_{10} (x/y + c)$ = log (x/y + c)

***

CHAPTER - 3 : MANAGING INPUT – OUTPUT STATEMENTS

C provides a library of functions. This library is called a **standard input output library**. It is denoted by **stdio**. The header file containing such library functions is called **stdio.h**.

There are two types of input-output functions. They are :

1. Formatted I/O functions
2. Unformatted I/O functions

FORMATTED OUTPUT STATEMENT

C provides the printf( ) function to display the data on the monitor. The printf( ) is included in stdio.h. It is used to display the results and messages on the screen. The general form of printf( ) statement is :

printf("control string", varlist);

where,

control string – specifies the type and format of the values to be displayed.

varlist – a list of variables to be displayed.

Example :
```
printf("Programming is an Art");
printf("%d",sum);
printf(%f%f",p,q);
printf("\nSum=%d", sum);
```

The following are different **format descriptors**:

Conversion Character	Meaning
%d	Print & Read a decimal integer
%u	Print & Read a unsigned integer
%s	Print & Read a string
%f	Print & Read a floating point number
%e	Print & Read a exponential floating point number
%c	Print & Read a single character
%g	Print & Read a floating point number
%i	Print & Read a decimal or octal or hexadecimal
%x	Print & Read a hexadecimal number
%h	Print & Read a short integer number
%o	Print & Read octal integer number
%p	Print & Read pointer

FORMATTED INPUT STATEMENT

To read the values for the variables in a program from the keyboard, C provides a function called scanf( ). This is used to accept numeric, character and string type of data. The address operator & (ampersand) is used to locate the values into variable memory.

Syntax :

scanf("control string", address_list);

where,

control string – It is a sequence of one or more character group. Each character is a combination of % symbol and a conversion character.

address_list – Address of memory locations where the values of input variables should be stored.

Example :
```
scanf("%d",&num);
scanf("%d%f%c",&a,&b,&c);
scanf("%d%s",&number, name);
```

Program to add two numbers.

```
#include<stdio.h>
#include<conio.h>
main()
{
int a,b,sum;
clrscr();
printf("Enter two numbers\n");
scanf("%d%d",&a,&b);
sum=a+b
printf("\nSum = %d",sum);
getch();
}
```

Program to accept three numbers and compute their sum and average.

```
#include<stdio.h>
#include<conio.h>
main()
{
int num1,num2,num3,sum;
float average;
clrscr();
printf("Enter three numbers\n");
scanf("%d%d%d",&num1,&num2,&num3);
sum=num1+num2+num3;
average=sum/3;
printf("\nThe Sum = %d",sum);
printf("\nThe Average = %f",average);
getch();
}
```

Program to find the simple interest :

```
#include<stdio.h>
#include<conio.h>
main()
{
int p,t,r;
float si;
clrscr();
printf("\nEnter the Principal :");
scanf("%d",&p);
printf("\nEnter the Term :");
scanf("%d",&t);
printf("\nEnter the Rate :");
scanf("%d",&r);
si=(p*t*r)/100;
printf("\nThe Simple Interest = %f",si);
getch();
}
```

Program to accept the temperature in Fahrenheit and convert it into Celsius.

```
#include<stdio.h>
#include<conio.h>
main()
{
float ct, ft;
clrscr();
printf("Enter the temperature in Fahrenheit\n");
scanf("%f",&ft);
ct=(ft-32.0)/1.8;
printf("Fahrenheit temperature = %6.2f\n", ft);
printf("Celsius temperature = %6.2f\n", ct);
getch();
}
```

UNFORMATTED INPUT FUNCTIONS

These functions are primarily concerned with reading the character type data from the keyboard. The getchar( ) and get( ) functions are used for this purpose. Since, they are included in the stdio.h, the C programs that use these functions should exclusively have the following preprocessor statement.

#include<stdio.h>

THE getchar( ) FUNCTION

This function reads a single character from the standard input device. There is no parameter within the parentheses. Its syntax is as follows :

ch_var = getchar( );

Where, ch_var is a character type variable to which an accepted character is assigned.

Example :

```
void main( )
{
char letter;
letter = getchar( );
printf("Your character = %c", letter);
}
```

THE gets( ) FUNCTION

This function reads in everything you enter from the keyboard until the ENTER key or RETURN is pressed. Here, everything means a string which is a sequence of all printable ASCII characters. The RETURN key that you pressed will not be stored at the end of the string. It overcomes the limitation of the scanf( ) statement with %s option. The syntax of gets( ) is

gets(string);

Where, string → is a sequence of characters and it is of type char.

Example :

```
void main( )
{
        char name[25];
        printf("Enter your name\n");
        gets(name);
        printf("Your Name = %s", name);
}
```

UNFORMATTED OUTPUT FUNCTIONS

These functions are mainly concerned with displaying or printing the character type data on the monitor. The putchar( ) and puts( ) functions are used for this purpose. Both these functions are defined in the header file <stdio.h>

THE putchar( ) FUNCTION

This function prints a single character on the screen. The character to be displayer is of type char. Its syntax is as follows.

putchar(ch_var);

Where, ch_var → is a character variable which is enclosed within the parentheses.

Example :

```
void main( )
{
        char letter;
        letter = getchar( );
        putchar(letter);
}
```

THE puts( ) FUNCTION

This function prints a string of characters on the screen. The newline character that signals the end of the string will not be displayed. The syntax of puts( ) is as follows:

puts(string);

Where, string → is a sequence of characters.

Example :

```
void main( )
{
        char message[20];
        printf("Enter the message to motivate students\n");
        gets(message);
        puts(message);
}
```

****

CHAPTER – 4 : COMPUTER ELECTRONICS

Introduction :

Early computer systems used electrical switches and when electrical switches were replaced by less mechanical devices such as vacuum tubes, than the transistor, the integrated circuit, the concept of switching on and off remained with computers but a representation of the on/off behavior of computers had to be made.

The term, digital, in computing and electronics applies to converting real world information to binary numeric form. The binary (base 2) number system represents two discreet values using two symbols or digits i.e., 0 and 1. The binary number system, where a zero symbolizes no electrical current (OFF) and one represents electrical current exists (ON).

All computer data (alpha-numeric, symbols and characters, audio, graphics and video) are represented or encoded using sequences of binary digits that are interpreted according to appropriate software. Computers are made up of electronic devices and electronic device can exist or either in ON or OFF state. For our convenience an ON State is represented by the code '1' and OFF state is represented by the code'0. They are called 'bits', an abbreviation for Binary Digit. The numbers represented by bits are known as binary numbers.

Basically number systems are classified in two types :

1) Non – Positional Number System
2) Positional Number System

Non-Positional Number System :

In this number system any number can be represented by arranging symbols in various positions. Each symbol represents a definite value irrespective of the position in which they appear.
For example : Roman Number System

Positional Number System :

In a positional number system, a number is represented by a set of symbols. Each symbol represents a particular value, depending on its position. The actual number of symbols used in a position system depends on its base.

Base :

The number of digits or basic symbols used in a positional number system is known as the BASE or Radix of the system.

1. Decimal Number System (Base 10) :

It is a positional number system with base 10 and thus it uses 10 symbols i.e., 0 to 9. Any number can be represented by arranging symbols in various positions. In the decimal system, each position represents a specific power of 10.

For example : The decimal number 654.52, written as $(654.52)_{10}$ or $654.52_{(10)}$ to specify base 10 is represented as follows :

Increasing Powers of 10				Decreasing Powers of 10	
	Hundreds	Tens	Units	One Tenth	One Hundredth
Weights	10^2	10^1	10^0	10^{-1}	10^{-2}
Digits	6	5	4	5	2

Thus, the expanded notation of

$$654.52_{(10)} = 6 \text{ X } 10^2 + 5 \text{ X } 10^1 + 4 \text{ X } 10^0 + 5 \text{ X } 10^{-1} + 2 \text{ X } 10^{-2}$$
$$= 6 \text{ X } 100 + 5 \text{ X } 10 + 4 \text{ X } 1 + 5 \text{ X } 0.1 + 2 \text{ X } 0.01$$
$$= 600 + 50 + 4 + 0.5 + 0.02$$
$$= 654.52$$

Note : In a positional number system all bits or digits to the left of the decimal or binary point have weights that are positive powers of base and those to the right have weights that are negative powers of base. The base is also called *Radix* and fractional point is called as *Radix point*.

2. Binary Number System (Base 2) :

The binary system is a positional system to the base 2. It uses two symbols 0 and 1. Each position represents specific power of 2.

For example : The binary number 1 1 0 1 . 1 1 written as $1\ 1\ 0\ 1\ .\ 1\ 1_{(2)}$ or $(1\ 1\ 0\ 1\ .\ 1\ 1)_2$ to specify base 2, is represented as follows :

Increasing Powers of 2					Decreasing Powers of 2	
Weights	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}
Digits	1	1	0	1	1	1

Thus, the expanded notation of $1\ 1\ 0\ 1\ .\ 1\ 1_{(2)}$

$$= 1 \text{ X } 2^3 + 1 \text{ X } 2^2 + 0 \text{ X } 2^1 + 1 \text{ X } 2^0 + 1 \text{ X } 2^{-1} + 1 \text{ X } 2^{-0}$$

3. Octal Number System (Base 8) :

The octal number system is a positional number system with base 8. It uses 8 symbols i.e., 0 to 7. In octal number system, each digit position corresponding to a power of 8.

For example : The octal number 43.12 written as $(43.12)_8$ or $43.12_{(8)}$ to specify base 8, is represented as follows :

Increasing Powers of 8			Decreasing Powers of 8	
Weights	8^1	8^0	8^{-1}	8^{-2}
Digits	4	3	1	2

Thus, the expanded notation of $43.12_{(8)}$

$$= 4 \text{ X } 8^1 + 3 \text{ X } 8^0 + 1 \text{ X } 8^{-1} + 2 \text{ X } 8^{-0}$$

4. Hexadecimal Number System (Base 16) :

The hexadecimal number system is a positional number system to the base 16. It uses 16 symbols to represent any number. The first 10 symbols are represented by digits 0 to 9 and the remaining 6 symbols by the letters A to F, representing the decimal values 10 to 15 respectively. Each position represents a specific powers of 16.

For example, the hexadecimal number BA85.12 is written as $(BA85.12)_{16}$ or $BA85.12_{(16)}$, is represented as follows :

Increasing Powers of 16					Decreasing Powers of 16	
Weights	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}
Digits	B	A	8	5	1	2

Thus, the expanded notation of $BA85.12_{(16)}$

$= B \times 16^3 + A \times 16^2 + 8 \times 16^1 + 5 \times 16^0 + 1 \times 16^{-1} + 2 \times 16^{-2}$

INTER CONVERSION OF NUMBER FROM ONE SYSTEM TO ANOTHER

INTEGER PART

1. Divide the integer part of the decimal number by the base 'b' of the new system. The remainder will give the right most digit of the integer part of the new number.
2. Divide the quotient again by the base 'b'. The remainder is the next digit from right.
3. Repeat step 2 until a zero quotient is obtained. Last remainder is the left most digit of the new number.

FRACTIONAL PART

1. Multiply the fractional part of the decimal number by the base 'b' of the new system. The integer part of the product gives the left most digit of the fractional part of the new number.
2. Multiply the fractional part of the product by the base 'b'. The integer part of the resultant product is the next digit from left.
3. Repeat step 2 until a zero fractional part or a repeated fractional part or a non-terminating fractional part occurs.

1. Decimal to Binary Conversion :

1) $25.625_{(10)} = ?_{(2)}$

				Product	Integer	Fractional-Part
2	2 5					
2	1 2	1	↑	0.625 X 2	1	0.250
2	6	0		0.250 X 2	0	0.500
2	3	0		0.500 X 2	1	0.000
2	1	1				
2	0	1				

Therefore, $25_{(10)} = 1\ 1\ 0\ 0\ 1$ & $0.625_{(10)} = 1\ 0\ 1$

Thus, the result is $25.625_{(10)} = 1\ 1\ 0\ 0\ 1\ .\ 1\ 0\ 1_{(2)}$

2. Binary to Decimal Conversion :

1) $1\ 1\ .\ 1\ 0\ 1\ 1_{(2)} = ?_{(10)}$

$1\ 1\ .\ 1\ 0\ 1\ 1_{(2)} = 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$

$= 1 \times 2 + 1 \times 1 + 1 \times 0.5 + 0 \times 0.25 + 1 \times 0.125 + 1 \times 0.0625$

$= 2 + 1 + 0.5 + 0 + 0.125 + 0.0625$

$= 3.6575_{(10)}$

3. Binary to Octal Conversion :

1) $0\ 1\ 1\ 1\ 0\ 1_{(2)} = ?_{(8)}$

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

Note : Place the 3 binary bits into one group from right side and write equivalent octal digit.

$\underline{0\ 1\ 1}\quad \underline{1\ 0\ 1}$

3 5

Thus, the result is $0\ 1\ 1\ 1\ 0\ 1_{(2)} = 3\ 5_{(8)}$

4. Binary to Hexadecimal Conversion :

1) $01101101_{(2)} = ?_{(16)}$

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

Note : Place the 4 binary bits into one group from right side and write equivalent Hexadecimal digit.

0 1 1 0 1 1 0 1

6 D

Thus, the result is $0\ 1\ 1\ 0\ 1\ 1\ 0\ 1_{(2)} = 6\ D_{(16)}$

5. Octal to Decimal Conversion :

1) $43.12_{(8)} = ?_{(10)}$

$43.12_{(8)} = 4 \times 8^1 + 3 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2}$

$= 4 \times 8 + 3 \times 1 + 1 \times 0.125 + 2 \times 0.015625$

$= 32 + 3 + 0.125 + 0.03125$

$= 35.15625_{(10)}$

6. Octal to Binary Conversion :

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

Note : Place the 3 bit binary equivalent of each digit below the number

1) $67.35_{(8)} = ?_{(2)}$

6 7 . 3 5

110 111 . 011 101

Thus the result is $67.35_{(8)} = 110111.011101_{(2)}$

7. Octal to Hexadecimal Conversion :

(a) First Covert Octal to Decimal

$43.12_{(8)} = ?_{(10)}$

$= 4 \text{ X } 8^1 + 3 \text{ X } 8^0 + 1 \text{ X } 8^{-1} + 2 \text{ X } 8^{-2}$

$= 4 \text{ X } 8 + 3 \text{ X } 1 + 1 \text{ X } 0.125 + 2 \text{ X } 0.015625$

$= 32 + 3 + 0.125 + 0.031250$

$= 35.15625_{(10)}$

(b) Then Convert Decimal to Hexadecimal

$35.15625_{(10)} = ?_{(16)}$

16	3 5			Product	Integer	Fractional-Part
16	2	3 ↑	0.15625 X 16	2.500	2 ↓	0.500
	0	2 ↑	0.50000 X 16	8.000	0 ↓	0.000

Therefore, $35_{(10)} = 2\ 3$ & $0.15625_{(10)} = 2\ 0$

Thus, the result is $35.15625_{(10)} = 2\ 3\ .\ 2\ 0_{(16)}$

Representation of Signed Numbers

In Binary System, we represent the sign of a number using an extra bit at the extreme right of the number and is called 'sign' bit. By convention the symbol '0' is used to represent the (+) sign and '1' to represent (−) sign.

In the case of binary numbers used in computers, the most significant bit represents the sign and the remaining bits the magnitude of the number. For example a 6 bit binary equivalent of a decimal number -15 is represented by $(1\ 0\ 1\ 1\ 1\ 1)_2$ and + 15 is represented by $(0\ 0\ 1\ 1\ 1\ 1)_2$.

This method of representing is known as sign magnitude representation.

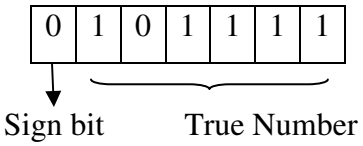
Representation of Signed numbers are :

1) By sign magnitude representation
2) By using 1's complement
3) By using 2's complement

Sign and Magnitude Representation

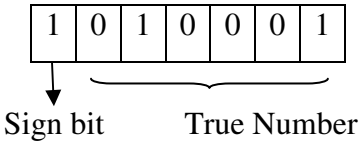
To represent Positive numbers, the magnitude is represented in the true binary form, and a sign bit 0 is placed in front of the MSB.

For example + 47 represented as follows :



To represent negative number, the magnitude is represented into 2's complement form, and a sign bit of 1 is placed in front of the MSB.

For example + 47 represented as follows :



1's Complement

The 1's complement of a binary number is obtained by complementary all the bits i.e., by changing 1's to 0's.

Example : Find the 1's complement of 1 0 1 1 0 0

Step1 :1 0 1 1 0 0 given number

Step2 : 0 1 0 0 1 1 complement each bit to form 1's complement.

Thus, 1's complement of 1 0 1 1 0 0 = 0 1 0 0 1 1

2's Complement

The 2's complement of a binary number is obtained by adding 1 to 1's complement of a binary number.

Example : Find the 2's complement of 1 0 1 1 0 0

Step1 :1 0 1 1 0 0 given number

Step2 : 0 1 0 0 1 1 complement each bit to form 1's complement.

\+ 1

Step3 : 0 1 0 1 0 0 is 2's complement of original number.

Binary Arithmetic :

Binary Arithmetic is a fundamental code for all digital computers and most other digital systems. It can perform all arithmetic operations like addition, subtraction, multiplication and division.

Binary Addition :

It is performed in the same manner as decimal addition. The following table shows that four basic rules for binary addition.

A		B	Sum	Carry
0	+	0	0	0
0	+	1	1	0
1	+	0	1	0
1	+	1	0	1

Examples :

```
   1 0 1          1 0 1 1 . 1 0 1
+  0 1 1      +   1 0 1 0 . 1 0 0
-------        -------------------
 1 0 0 0        1 0 1 1 0 . 0 0 1
-------        -------------------
```

Binary Substraction :

Binary subtraction rules are as follows :

A		B	Sum	Carry
0	-	0	0	0
1	-	0	1	0
1	-	1	0	0
0	-	1	1	1

Examples :

```
   1 0 1          1 0 0 1 0 1 0
-  0 1 1      -       1 0 1 0 1
-------        -------------------
   0 1 0        0 1 1 0 1 0 1
-------        -------------------
```

Subtraction of Binary number using 1's Complement :

Case 1: Substraction of Smaller number from Larger number

1. Determine 1's Complement of smaller number
2. Add 1's compliment to the larger number
3. We get carry. Addition end around carry to the above gives the result of subtraction.

Example :

Subtract 1001 – 1000 using 1's Complement

1. 1'st complement of 1000 is 0111
2. Add 1's complement to the larger number

```
  1 0 0 1
  0 1 1 1
-----------
  1 0 0 0 0
-----------
```

3. Add carry 1 to the above result

```
  0 0 0 0
+       1
-----------
  0 0 0 1
-----------
```

Therefore, 1 0 0 1 – 1 0 0 0 = 0 0 0 1 (9 – 8 = 1)

Case 2: Substraction of Larger number from Smaller number

1. Determine 1's Complement of larger number
2. Add 1's compliment to the smaller number
3. There will be no carry. To get answer take the 1's complement of the result and put negative sign.

Example :

Subtract 111 – 101 using 1's Complement

1. 1's complement of larger number 111 is 000
2. Add 1's complement to the smaller number

```
    1 0 1
+   0 0 0
  -------
    1 0 1
  -------
```

3. Find 1's complement of 101 and put negative sign

   i.e., - 0 1 0

Therefore, 1 0 1 – 1 1 1 = - 0 1 0 (5 – 7 = -2)

Subtraction of Binary number using 2's Complement :

Case 1: Substraction of Smaller number from Larger number

1. Determine 2's Complement of smaller number
2. Add 2's compliment to the larger number
3. We get carry. Neglect the carry and the remaining gives the result of subtraction

Example :

Subtract 1001 – 1000 using 2's Complement

1. 2's complement of 1000 is

```
    0 1 1 1
+       1
  ---------
    1 0 0 0
  ---------
```

2. Add 2's complement to the larger number

```
    1 0 0 1
+   1 0 0 0
  -------
  1 0 0 0 1
  -------
```

3. Neglect the carry i.e., 0 0 0 1

Therefore, 1 0 0 1 – 1 0 0 0 = 0 0 0 1 (9 – 8 = 1)

Case 2: Substraction of Larger number from Smaller number

1. Determine 2's Complement of larger number
2. Add 2's compliment to the smaller number
3. There will be no carry. To get answer take the 2's complement of the result and put negative sign.

Example :

Subtract 111 – 101 using 1's Complement

1. 2's complement of larger number 111 is

   i.e., 0 0 0

   \+ 1

   ---

   0 0 1

   ---

2. Add 2's complement to the smaller number

   1 0 1

   \+ 0 0 1

   ---

   1 1 0

   ---

3. Find 2's complement of 110 and put negative sign

   i.e., 0 0 1

   \+ 1

   ---

   0 1 0

   ---

   i.e., - 0 1 0

Therefore, 1 0 1 – 1 1 1 = - 0 1 0 (5 – 7 = -2)

Computer Codes :

Computer understands everything only in binary. Therefore, when we input numbers, alphabets and other special symbols, they must be represented in the binary format. There are three such coding standards. There are BCD, ASCII and EBCDIC.

1. BCD (Binary Coded Decimal System :

In BCD each digit of a decimal number is independently converted to 4-bit binary number. For example the decimal number 573 would be represented in the 4 bit BCD code as

0 1 0 1 0 1 1 1 0 0 1 1

5 7 3

BCD Coding system is used to represent only decimal numbers, 4-bits are insufficient to represent the various characters used by computer. Therefore 6-bit BCD Code was developed. In this code two more bits called as 'zero position' are added. It is possible to represent 64 code groups i.e., 10 decimal digits, 26 alphabets and 28 special characters using 6 bit code.

2. ASCII (American Standard Code for Information Interchange :

It is most widely used alphanumeric code for printers, keyboards and terminals which are interface with computer to represent data. This is 7 bit code and also it has 128 possible code groups i.e., alphabets, numbers, special characters and control character (Enter Key, Escape Key, Space bar etc.,). A is represented in ASCII as 1000001 whose decimal equivalent is 65.

3. EBCDIC (Extended Binary Coded Decimal Interchange Code) :

It is also an alphanumeric code used in IBM computers and mainframe applications. It is an 8 bit code representing 256 different code group. A is represented in EBCDIC code as 11000001.

Digital Logic :

The word Logic is used to describe the circuits which can duplicate specific function of decision making performed by the human mind. In the logic relevant to computers we know that there can be only two states. This logic is thus called two state logic or bivalent logic. Such a logical method was developed by ARISTOTAL for getting at the truth. The method was subsequently developed by mathematician DE-MORGAN and GEORGE BOOLE into a very powerful mathematical tool.

A Gate is a simple electronic circuit or device that performs logical functions. It has one or more inputs and output. Gates are called binary logic gates 1 and 0 are inputs and outputs.

Truth Table

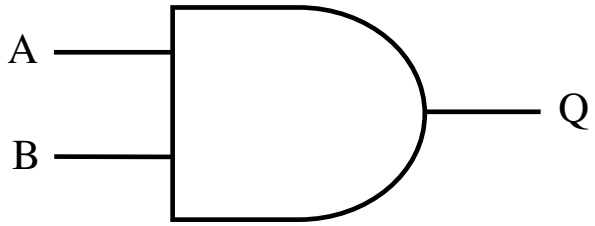
Truth Table is a table which shows all inputs and outputs possibilities of a logical circuits or gate.

Types of Logic Gates

1. AND Gate
2. OR Gate
3. NOT Gate
4. NAND Gate
5. NOR Gate

AND Gate

This is an electronic decision making element with one or more inputs and single outputs. Its function is to implement the AND operation (i.e., Logical Multiplication). The logical symbol for AND Gate is



Two inputs AND Gate truth table

Input A	Input B	Output Q=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Three inputs AND Gate truth table

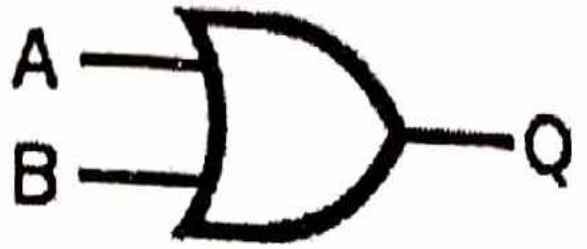
Input A	Input B	Input C	Output Q=A.B.C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

AND Gate Rule

1. Logic 1 when all inputs are 1
2. Logic 0 when any input is 0

OR Gate

This is an electronic decision making element with one or more inputs and single output. Its function is to implement the OR operation (i.e., Logical Addition). The logical symbol for OR Gate is



Two inputs OR Gate truth table

Input A	Input B	Output Q=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Three inputs OR Gate truth table

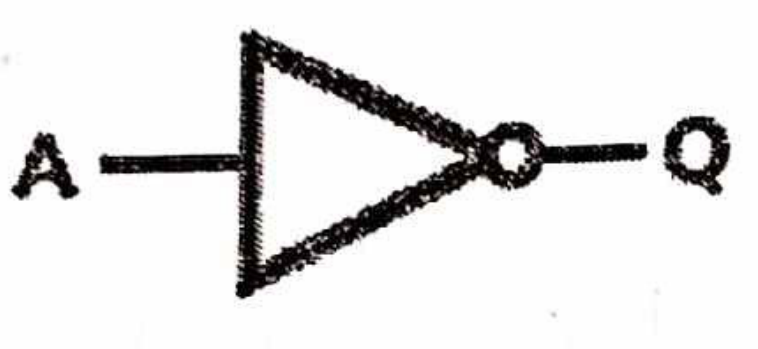
Input A	Input B	Input C	Output Q=A+B+C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

OR Gate Rule

1. Logic 1 when an inputs is 1
2. Logic 0 when all inputs are 0

NOT Gate

This is an electronic decision making element with one input and one output. Its function is to implement the NOT operation (i.e., Inversion or Logical Complement) . The logical symbol for NOT Gate is



Truth table of NOT Gate is

Two inputs OR Gate truth table

Input A	Output Q (Complement of A)
0	1
1	0

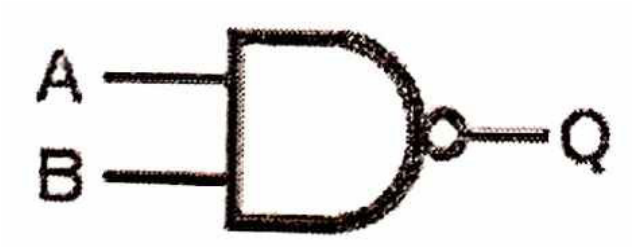
NOT Gate Rule is

1. Logic 1 when input is 0
2. Logic 0 when input is 1

NAND Gate

This is combination of NOT and AND Gates. This is like AND Gate but with the output complimented.

The logical symbol for NAND Gate is



Two inputs NAND Gate truth table

Input A	Input B	A.B	Output Q=NOT(A.B)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

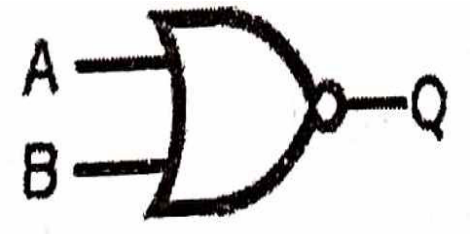
NAND Gate rule is

1. Logic 1 when any input is 0
2. Logic 0 when all inputs are 1

NOR Gate

This is combination of NOT and OR Gates. This is like OR Gate but with the output complimented.

The logical symbol for NOR Gate is



Two inputs NOR Gate truth table

Input A	Input B	A+B	Output Q=NOT(A+B)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NOR Gate rule is

1. Logic 1 when all inputs are 0
2. Logic 0 when any inputs is 1

Universal Gates

NAND and NOR gates are called universal gates because the other gates (i.e., AND, OR and NOT) can realized these individual gates.

****

CHAPTER – 5 : INTERNET TOOLS

CONCEPT OF NETWORKING

Exchange of information is required in many situations between two users. In such situations, data communications is very much required. Data communication refers to the transmission of data from one computer to another by a data carrier cable that connects two computers.

Computer Network can be defined as an interconnected of autonomous (independent) computers. Normally, these computers are connected through a copper wire. Computer network connection can be of two types.

There are two important computers in a network i.e., Server Computer and Client Computer. Server Computer is the main computer that stores the data of all the clients, shares the files and peripherals with the clients. Client Computer is a subordinate computer connected to the Server Computer. It can access the files and peripherals of Server Computer. There can be more than one client computers.

THE USES OF A NETWORK

The computer networks offer many uses to its users. While computer networks provide data sharing, device sharing, flexibility and high reliability in data communication, they do offer very specific benefits to the network users. They are :

1] Simultaneous Access :

Because of networked environment, all users in a company can access the same data at the same time. A central server or simply server maintains the commonly used data which can be shared among several people. If one user modifies the shared data, other users in the network will see the modification when they access it.

2] Shared Peripheral devices :

It is very difficult to provide every user one PC and one printer in a company. Because of the networking of PCs, a single printer can be made as a shared device. All users in that network can share this printer to print their jobs. This reduces the cost and space required to purchase and install several printers.

3] Personal Communication :

Because of computer networks and the Internet which is a network of networks, it is possible for us to have personal communications with any person in the world. The common form of personal communication is achieved through an electronic mail. However, due to many protocols and electronic equipment, we can have teleconference with the remote user. Video conferencing and audio conferencing have made the world very small.

4] Easier to backup :

Data in any business is extremely critical. One should take the back up of such data. Network provides a shared storage device on which the backups of critical data is stored and accessed as and when required.

COMMON TYPES OF NETWORKS

1] LOCAL AREA NETWORKS [LAN]

This is an interconnection of autonomous computers within a single building or a small campus. The range of LANs may vary from 10m to 1.5 km. LANs are small sized networks. In LAN transmission technology, each computer usually contains a network interface device that connects the computer directly to the network medium such as a copper wire or coaxial cable. LANs operate between 10 Mbps and 2 Gbps. Since LAN technology covers short distances, they offer lower delays and make very few errors.

2] WIDE AREA NETWORKS [WAN]

This is a collection of computers that are geographically spread over a wide area. There is no limitation on the distance it spans. A WAN can span over a country or a continent. WANs operate usually at slower speeds. Typical speeds for a WAN range from 2 Mbps to 625 Mbps. They offer a much greater delay between connections. It is due to far distances. This delay is due to sending signals to a satellite orbiting the earth.

3] METROPOLITAN AREA NETWORK [MAN]

This is a larger computer network as compared to LANs. It connects multiple corporate LANs together. It is generally called MAN. You might have heard people saying "There is a MAN between a LAN and WAN". The MANs are high-speed networks for sharing the regional information and resources.

DIFFERRENT BETWEEN LAN AND WAN :

	LAN	WAN
1	Diameter of not more than a few kilometer	Span entire countries
2	A total rate of at least several mbps	Data rate less than 1 mbps to 10 mpbs
3	Complete ownership by single organization	Owned by multiple organization
4	Very low error rates	Comparatively higher error rates

INTERNET

Internet is a network of many computer networks. It connects LANs, WANs and even our own personal computer. It is also called the Information Super Highway. We can access any information from any place at anytime on the Internet. We can download to our PC the research papers, send mails to the friends and relatives, watch a movie, deposit or withdraw money online and sell or buy products and services.

- The internet is often referred to as the "Information Super Highway"
- No one owns the Internet, but everyone connected to it owns part of it.
- It is not governed by any central authority but there are agencies and societies that perform coordinating functions, such as the registration of domain names.
- In 1988, there were about 1 million people with access to the Internet. Currently, there are more than 500 million Internet connections, divided across almost every country in the world.
- The web sites produced by institutions, business and private users cover just about every imaginable topic.
- The Internet was developed in 1969 as a communications network for the US military. Soon, other government agencies, colleges and research companies began using the Internet.
- Internet telephony is a relatively new technology used to digitize sound and transfer it across the Internet.
- Video conferencing technology for the World Wide Web had been developed.

In order to avail the facilities of internet, our PCs must be connected to the Internet. Therefore, we have to become the subscriber to the Internet. There are many Internet service providers around the globe. In India, BSNL (Bharatiya Sanchar Nigama Ltd) provides Internet Services. Presently, some private organizations like Satyam Infoway, BPL, Wipro etc., started providing Internet Services. These organizations are called ISPs (Internet Service Providers).

CONNECTING TO THE INTERNET :

To connect your PC to the Internet, the following hardware and software are required.

1. Modem : It stands for Modulator to Demodulator. It is used to connect the PC to a telephone line.
2. Telephone Line : To transmit data or message over a wide area.
3. A PPP (point to point protocol) account with an Internet service provider.
4. Internet connection software (TCP/IP)
5. Software for E-mailing, browsing the web and so on. (Internet Explorer)

SERVICES OF INTERNET :

1] Net Banking :

Net Banking as replaced the conventional way of banking. Net Banking help us in depositing, withdraw or updating the account with just a click on mouse. We can also get the required information about our Bank Account.

2] Education :

No more requires the student has to go college, register and attend the class. Infact a student can now not only register and attend the classes but also give exams for that particular class using Internet.

3] Employment :

This system with which job sector can register and obtain information for the vacancies with the various companies.

4] Seminar, Discussion :

Participating in a discussion about your favourite TV Show with like minded people across the globe. The Internet community is an open form that no government or restrictions. The citizens of this community are termed as Netizens.

5] Greetings :

Sending and receiving greetings for various occasions across the globe.

6] Searching for Information :

Internet provides search engine websites. These websites allows to search the information using keywords. The popular search engines are

www.google.com, ww.khoj.com, www.yahoo.com etc.,

7] Chatting :

If you like to speak or talk and makes new friends. Then Internet is the best place, with chat program you can talk with a group of people whose geographical location you don't know.

8] E-Mails :

With E-Mails we can send and receive mail messages. E-Mails can be send globally just by paying local telephone charges.

9] Telnet to Other Computer :

With Telnet program you can connect to another computer and use as if you are sitting at it's keyboard.

INTRANET AND EXTRANETS

Intranet is a scaled-down Internet. It is a corporate specific Internet. That is, each and every company may have its own private Internet to serve the needs of its employees and selected contractors who work within the company. Outsiders cannot access intranet. It connects a set of computers using a standard Internet Protocols such as TCP/IP and HTTP.

Extranet is also a scaled-down Internet. It is a great facility to the employees who can access the resources of their company while they are travelling onsite or working from home. It plays a role of business-to-business Internet which allows a limited, controlled, secure access between a company's intranet and designated, authenticated users located outside the organization. The employees or business partners who have a valid username and password can access the selected information.

DEFINITION OF WEBPAGE

A hypertext document is called a web page. Web pages are just ordinary computer files with a few tags added to tell your computer how to display the contents of the file. A webpage is a page within a website. A website contains few or many of such web pages.

MEANING OF HTML

HTML stands for Hypertext Markup Language. HTML is a language used to markup the web contents. That is, it specifies how the contents of the WebPages are displayed on the browser. It stands for hypertext markup language. HTML provides a set of tags (or commands) for marking up the web contents.

DEFINITION OF WEBSITE

It is collection of web pages. We can also say that it is a computer system that is set up to distribute documents stored in its database. Websites range in size from as little as one page to a vast number of pages, such as those of search engine's database or a full text book. The websites are hoisted on the web servers.

WEB ADDRESSES (URL)

URL is an acronym for Uniform Resource Locator. The URL specifies the Internet address of a file stored on a host computer connected to the Internet. Every file on the Internet, no matter what its access protocol, has a unique URL. Web browsers use the URL to retrieve the file from the host computer and the specific directory in which it resides. This file is downloaded to the user's client computer and displayed on the monitor connected to the machine.

URLs are translated into numeric addresses using the Domain Name System (DNS). The DNS is a worldwide system of servers that stores location pointers to websites. The general format of a URL is as follows :

type://address/path/filename

Where,	type	→	type of server where the file is located
	address	→	address of the server
	path	→	location within the file structure of the server
	filename	→	name of the file which has the required contents

Example: http://library.standard.edu/research_help/index.html

Domain Names describe organizational or geographical relativities. It indicates the country where the network connection is in the kind of organization that owns it etc,

Following are examples of domain names and their meaning

com – Commercial
edu – Educational
gov – Government
net – An administrative organization for a network,
mil – Non classified Military Networks,
org – Usually private organization and other

There are also domain names for countries

dc – Germany (Dutch land)
it – Italy
nz – New Zealand
in – India

WEB BROWSING TOOLS

A web browser or simply browser is a program designed to find the hypertext documents on the web. Once it finds the documents it opens that documents on the computer from which search was made. The browser translates the web files into text, images, sound, video etc., Some of the popular Web browsers are :

- Internet Explorer(Microsoft)
- Netscape Navigator
- Netscape Communicator
- Lynx
- Mosaic
- Hot Java

ISP'S IN INDIA AND THEIR RESPONSIBILITIES

In order to avail the facilities of internet, our PCs must be connected to the Internet. Therefore, we have to become the subscriber to the Internet. There are many Internet service providers around the globe. In India, BSNL (Bharatiya Sanchar Nigama Ltd) provides Internet Services. Presently, some private organizations like Satyam Infoway, BPL, Wipro, TATA DOCOMO, Airtel, Idea etc., started providing Internet Services. These organizations are called ISPs (Internet Service Providers).

RESPONSIBILITIES OF ISP's

1) Provide a reliable and accessible conduit for traffic and services.
2) Provide authentic and authoritative routing information
3) Provide authentic and authoritative naming information
4) Report anonymzed security incident statistics to the public
5) Educate customers about threats
6) Inform customers of apparent infections in their infrastructure
7) Warn other ISPs of imminent danger and help in emergencies
8) Avoid aiding and abetting criminal activity.

E-MAIL

This is a short form of Electronic Mail. It allows users to compose messages and send them to remote individuals or groups. It also allows users to read the incoming e-mails. It is the most widely used Internet service. It offers a fast and convenient method of transferring messages. Everyone should have a e-mail address to send & receive e-mails.

An E-mail system consists of two subsystems : the user agents and message transfer agents. The user agents allow users to send and receive e-mail. The message transfer agents move the messages from one computer to another over network.

The E-mail system is no longer restricted to exchange only text information it is also used to deliver voice mail, facsimiles and audio-visual images. When you compose an e-mail and send it to the receiver, it will be available to the receiver within a few seconds. So, e-mailing is one of the fastest ways of communication. Every e-mail address has the form:

username@organization_name.typeoforganization

Example : shafisardarkpl@gmail.com

Advantages of E-mail :

1) **Low Cost** : Email extremely cost-effective way to move information around, the world.
2) **Speed** : Electronic mail can be delivered almost as fast as the wire can carry it.
3) **Waste reduction** : Email goes a long way toward reducing the wastage of paper in the modern office
4) **Ease of use** : It is easy to send an email. You don't have to retype it, find on envelope, get a stamp then find mail box
5) **Record maintenance** : Because all messages are digital files, you can automatically maintain a record of communications

CHATTING

On the Internet, chatting is talking to other people who are using the Internet at the same time you are. Usually, this "talking" is the exchange of typed-in messages requiring one site as the repository for the messages (or "chat site") and a group of users who take part from anywhere on the Internet. In some cases, a private chat can be arranged between two parties who meet initially in a group chat. Chats can be ongoing or scheduled for a particular time and duration. Most chats are focused on a particular topic of interest and some involve guest experts or famous people who "talk" to anyone joining the chat. A chat can also be conducted using sound or sound and video.

VIDEO CONFERENCING

A videoconference is a live connection between people in separate locations for the purpose of communication, usually involving audio and often text as well as video. At its simplest, videoconferencing provides transmission of static images and text between two locations. At its most sophisticated, it provides transmission of full-motion video images and high-quality audio between multiple locations.

A videoconference can be thought of as a phone call with pictures - Microsoft refers to that aspect of its NetMeeting package as a "web phone" - and indications suggest that videoconferencing will someday become the primary mode of distance communication.

SEARCH ENGINE

This is a program that locates information in its database. A search engine functions as a service that searches for information on the Internet. It responds by matching your query terms to the search engine's index terms in its database, ranking the matches and returning the hits to you. You can type your search keyword and submit for searching. Within a few seconds you will find lots of related pages displayed on the screen. If you do not find the exact match you can search further and so on till you get what you want. Some of the popular search engines are listed below.

- Alta Vista
- Dogpile
- Excite
- Fast
- Google
- HotBot
- Infoseek
- Look Smart
- Savvy Search
- Yahoo
- Snap
- WebCrawler
- Khoj

UPLOADING AND DOWNLOADING FILES

The process of copying a webpage onto a web server is called publishing the page. It is also called Uploading Files.

The reverse process of copying a webpage from the web server to your local disk is called Downloading Files.

****