HSE I Focus Area Notes Computer Science

Chapter 1: The Discipline of Computing

Abacus: Considered as the first computer for basic arithmetical calculations. Consist of beads on movable rods divided into two parts.

Difference Engine: In 1822, Babbage invented this machine that could perform arithmetic calculations and print results automatically.

Analytical Engine: The Engine had a 'Store' (memory) , a separate 'Mill' (processor) and input/output devices. Developed by Charles Babbage in 1833, the 'Father of Computer'. Agusta Ada King, the first programmer in the world wrote instructions for this machine.

Generations Of Computers:

Criteria	Generation				
	First	Second	Third	Fourth	Fifth
Technology	Vacuum Tube	Transistor	Integrated Circuit	Microprocessor	Artificial Intelligence
Operating System	None	None	Yes	Yes	Yes
Language	Machine	Assembly	High Level	High Level	High Level
Period	1940-1956	1956-1963	1964-1971	1971-Present	Present and Yet to come

Table 1.1 : Comparative features of five generations of computers

Chapter 2: Data Representation and Boolean Algebra

Number System

Table 2.1 shows the base and symbols used in different number systems:

Number System	Base	Symbols used
Binary	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Table 2.1 : Number systems with base and symbols

Number conversions: to convert the numbers of one base to the equivalent numbers in other bases.

Decimal to Binary:
Repeated division by 2 and grouping the remainders

Find binary equivalent of decimal number 25.

2 | 25		Remainders
2 | 12	1	LSB
2 | 6	0	
2 | 3	0	
2 | 1	1	
0	1	MSB

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

Decimal to Octal: Repeated division by 8 and grouping the remainders

Example: Find octal equivalent of decimal number 125.

		Remainders	
8	125		
8	15	5	LSD
8	1	7	
	0	1	MSD

$(125)_{10}=(157)_8$

Decimal to Hexadecimal: Repeated division by 16 and grouping the remainders

Example: Find hexadecimal equivalent of 380.

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

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

Binary to Decimal: Multiply binary digit by place value(power of 2) and find their sum

Example: Convert $(11011)_2$ to decimal.

$$(11011)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$$
$$= 16 + 8 + 2 + 1$$
$$= 27$$

$(11011)_2=(27)_{10}$

Octal to Decimal : Multiply octal digit by place value (power of 8) and find their sum

Example: Convert $(1005)_8$ to decimal.

$$(1005)_8 = 1\times8^3 + 0\times8^2 + 0\times8^1 + 5\times8^0$$
$$= 512+5$$
$$= 517$$

$(1005)_8=(517)_{10}$

Hexadecimal to Decimal : Multiply hexadecimal digit by place value (power of 16)and find their sum

Example: Convert $(2D5)_{16}$ to decimal.

$$(2D5)_{16} = 2\times16^2+13\times16^1+5\times16^0$$
$$= 512+208+5$$
$$= 725$$

$(2D5)_{16}=(725)_{10}$

Short Cut Method of Convertions:

Octal to binary conversion:

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

3-bit binary equivalent of each octal digits are

4	3	7
↓	↓	↓
100	011	111

$(437)_8=(100011111)_2$

Example: Convert $(7201)_8$ to binary.

Hexadecimal to binary conversion:

Example: Convert $(2F15)_{16}$ to binary.

4-bit binary equivalent of each hexadecimal digits are

2	F	1	5
↓	↓	↓	↓
0010	1111	0001	0101

$(2F15)_{16} = (10111100010101)_2$

Binary to octal conversion:

Example: Convert $(101100111)_2$ to octal.

We can group the given binary number 101100111 from right as shown below.

101	100	111
↓	↓	↓
5	4	7

$(101100111)_2 = (547)_8$

Binary to hexadecimal conversion:

Example: Convert $(101100111010)_2$ to hexadecimal.

We can group the given binary number 101100111010 from right as shown below.

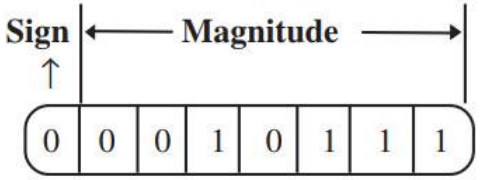
1011	0011	1010
↓	↓	↓
B	3	A

$(101100111010)_2 = (B3A)_{16}$

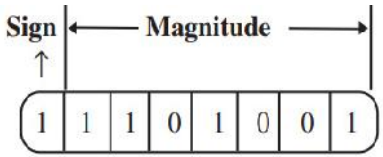
Representation of numbers(integers):

(i) Sign and magnitude representation: In this method, first bit from left (MSB) is used for representing sign of integer and remaining 7-bits are used for representing magnitude of integer. For negative integers sign bit is 1 and for positive integers sign bit is 0.

Example: Represent + 23 in sign and magnitude form



Example: Represent -- 23 in sign and magnitude form



ii) 1's complement representation : 1's complement of a binary number is obtained by replacing every 0 with 1 and every 1 with 0.

Binary Number	1's Complement
11001	00110
10101	01010

iii) 2's complement representation: If the number is positive 8-bit form binary itself is the representation. 2's complement of a binary number is calculated by adding 1 to its 1's complement.

Example: Represent -38 in 2's complement form.

Binary of 38 in 8-bit form $= (00100110)_2$

-38 in 2's complement form $= 11011001 + 1$

$= (11011010)_2$

Example: Represent +38 in 2's complement form.

Binary of 38 in 8-bit form $= (00100110)_2$

+38 in 2's complement form $= (00100110)_2$

(No need to find 2's compleme

Representation of characters:

ASCII: American Standard Code for Information Interchange uses 7 bits to represent each character in computer memory

UNICODE: It is a 16 bit or 32 bit code It is used to represent all the characters used in the written languages in the world.

Boolean operations: The operations performed on the Boolean values 0s and 1s. The operations are OR (Logical Addition), AND (Logical Multiplication), NOT (Logical Negation).

a)The OR operator and OR gate: performs logical addition operations and the symbol used for this is +.The expression A+B is read as A OR B.

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

Table 2.9 : Truth table of OR operation

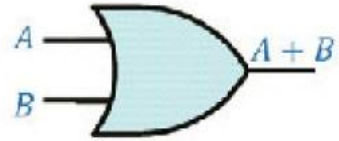


Fig. 2.5 : Logical OR gate

b)The AND operator and AND gate: performs logical multiplication operations and the symbol used for this is . (dot).The expression A.B is read as A AND B.

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

Table 2.12 : Truth table of AND operation

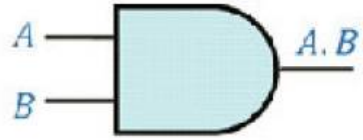


Fig. 2.8 : Logical AND gate

c)The NOT operator and NOT gate: The NOT operator performs logical negation and the symbol used for the operation is - (Over bar) or ' (dash) eg:A'

A	$\overline{A}$
0	1
1	0

Table 2.15 : Truth table of NOT operation

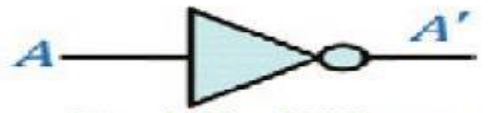


Fig. 2.11 : NOT gate

Circuit designing for Simple Boolean Expressions:

Example: Construct a logical circuit for Boolean expression $f(X, Y) = X.Y + \overline{Y}$

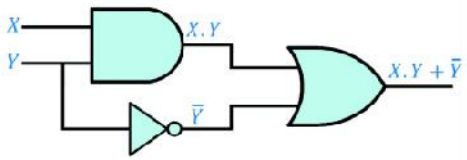


Fig 2.13 : f (X, Y) = X.Y + $\overline{Y}$

Example: Construct a logical expression for $f(a, b) = (a + b) . (\overline{a} + \overline{b})$

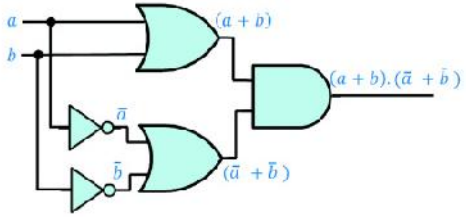


Fig 2.14 : f (a, b) = (a + b) . ($\overline{a}$ + $\overline{b}$)

Example: Construct logical circuit for the Boolean expression $\overline{a}$ .b + a.$\overline{b}$

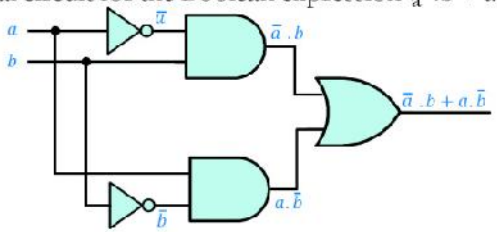


Fig 2.15 : f (a, b) = $\overline{a}$ .b + a.$\overline{b}$

Chapter 3:Components of the Computer System

Processor: The processor also known as CPU, is responsible for computing and decision making.

1)Accumulator : It stores the result of ALU operations.

2) Memory Address Register (MAR): It stores the address of a memory location to which data is either to be read or written.

3) Memory Buffer Register (MBR): It holds the data to be written to or read from the memory.It is also known as Memory Data Register(MDR).

4) Program Counter (PC): It holds the address of the next instruction to be executed by the CPU.

5) Instruction Register (IR): It Stores the current instruction.

Ports: Ports on the motherboard are used to connect external devices.

a. Serial port:A serial port / serial communication port transmits data one bit at a time.

b. Parallel port: Parallel ports can transmit several bits of data simultaneously.

c. USB port:USB (Universal Serial Bus) is a connection that provides high speed data communication between devices.

d. LAN port:(LAN) port is a port connection that allows a computer to connect to a network using a wired connection.

e. PS/2 port: Personal System/2 (PS/2) ports are special ports for connecting the keyboard and mouse

f**. Audio ports:**Audio ports are used to connect audio devices like speakers, microphone, etc.

g. Video Graphics Array (VGA) port: The VGA port is used to connect a monitor or a projector to a computer.

h. High Definition Multimedia Interface (HDMI):HDMI is a type of digital connection capable of transmitting high-definition video and multi channel audio over a single cable

Primary Memory: Primary memory holds data, intermediate results and results of ongoing jobs temporarily.

Random Access Memory (RAM): RAM is a volatile memory. Data or instructions to be processed by the CPU must be placed in the RAM.

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

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

e-Waste disposal methods:

a)Reuse: It refers to second-hand use or usage after the equipment has been upgraded or modified.

b)Incineration: It is a controlled and complete combustion process in which the waste is burned in specially designed incinerators at a high temperature

c)Recycling of e-Waste: Recycling is the process of making or manufacturing new products from a product that has originally served its purpose.

d)Land filling: e-Waste material is buried in soil making deep pits.

System software: is a set of system programs which aids in the execution of a general user's computational requirements on a computer system. Components of system software are Operating System,Language Processors and Utility Software

Operating system(OS): Operating system is a set of programs that acts as an interface between the user and computer hardware. Operating system controls and co-ordinates the operations of a computer.

Major functions of an operating system

i. Process management: It takes care of the allocation, de-allocation and scheduling of system processes.

ii. Memory management: It takes care of allocation and de allocation of memory.

iii. File management: It takes care of file related activities such as organising, naming, storing, retrieving, sharing, protection and recovery.

iv. Device management: It handles the devices connected to the computer by combining both hardware and software techniques.

Eg of operating systems are DOS, Windows, Unix, Linux, Mac OS, etc.

Language processors: Language processors are the system programs that translate programs written in high level language or assembly language into its equivalent machine language.

Types of language processors:

Assembler: Translates the program code written in assembly language to machine language

Compiler: Translates a program written in high level language into machine language in a single run. Compilers are used in C, C++ ....

Interpreter: Converts a HLL program into machine language line by line

Free and open source software: Free and open source software gives the user the freedom to use, copy, distribute, examine, change and improve the software.

The Free Software Foundation (FSF) defines the four freedoms for free and open source 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. Access to source code should be provided.

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, so that the whole community benefits.

Chapter 4 : Principles of Programming and Problem Solving

Phases in programming

1. Problem identification
2. Preparing algorithms and flow charts
3. Coding the program using programming language
4. Translation
5. Debugging
6. Execution and testing
7. Documentation

Debugging: programming errors are known as 'bugs' and the process of detecting and correcting these errors is called debugging.

1)Syntax errors : When the rules or syntax of the programming language are not followed. Such program errors typically involve incorrect punctuation, incorrect word sequence, undefined term, or illegal use of terms or constructs. the syntax errors will be detected and displayed during execution.

2)Logical error: is due to improper planning of the program's logic.

3)Run-time error:due to the inappropriate data that is encountered in an operation causes interruption in execution.

Example 3.2: To find the area and perimeter of a rectangle

We know that this problem can be solved, if the length and breadth of the rectangle are given. The result can be obtained by using the following formula:

Perimeter = 2 (Length + Breadth), Area = Length × Breadth

Let L and B be variables for length and breadth; and P, A be variables for perimeter and area.

Step 1: Start
Step 2: Input L, B
Step 3: P = 2 * (L + B)
Step 4: A = L * B
Step 5: Print P, A
Step 6: Stop

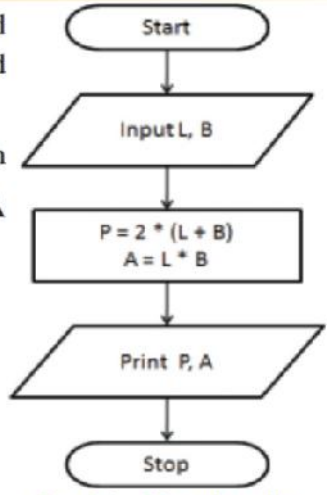


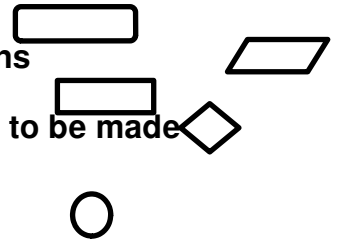
Fig. 3.6 : Flow chart for Area and Perimeter

Algorithm: an algorithm may be defined as a finite sequence of instructions to solve a problem.

Flowchart: pictorial representation of an algorithm

Flow Chart Symbols:

1.Terminal:-indicate start and stop
2.Input/output: denotes input/output functions
3.Process: represents processing steps
4.Decision:indicate the point decisions have to be made
5.Flow lines: indicate the flow of operation
6.Connector:Used to join flow of lines



Eg.Algorithm and Flow Chart to find biggest of two

Step 1: Start
Step 2: Input H1, H2
Step 3: If H1 > H2 Then
Step 4: Print H1
Step 5: Else
Step 6: Print H2
Step 7: End of If
Step 8: Stop

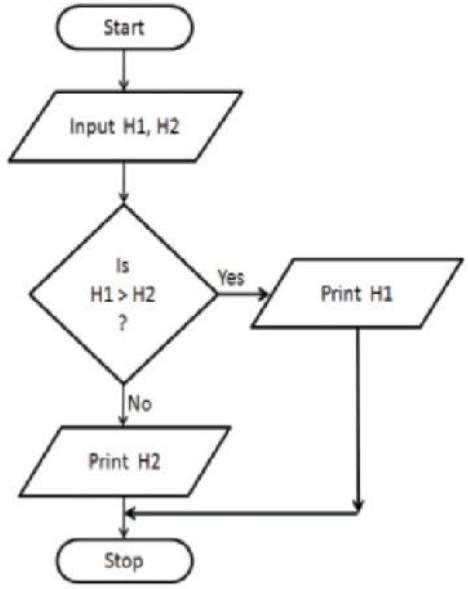


Fig. 3.8 : Flowchart to find larger value

Chapter 5 : Introduction to C++ Programming

Tokens: The fundamental building blocks of the program.

a)Keywords.: The words (tokens) that convey a specific meaning to the language compiler are called keywords. **eg.**asm , continue , float ,new ,signed.

b)Identifiers: Identifiers are the user-defined words that are used to name different program elements such as memory locations, statements, functions, objects, classes etc.**eg.** Count, Sumof2numbers, _1stRank,Main, Average_Height, FOR.

The rules for naming identifiers:

- The name can start with a letter. It can start with underscore (_) also.
- Keywords cannot be used as identifier.
- White spaces and special symbols are not allowed, except underscore(_).
- Upper case and lower case letters are treated differently ie, case sensitive

c)Literals:the type of tokens called literals to represent data items that never change their value during the program run. They are often referred to as constants. Literals can be divided into four types as follows:

1. Integer literals :are whole numbers without fractional part eg: 9856, 569…

2. Floating point literals:numbers having fractional parts eg 3.14, 7.06

3. Character literals:single characters enclosed within single quotes eg:'A', 'y'

4. String literals:enclosed in double quotes eg:"A", "Hello"

d)Punctuators: Special symbols that have syntactic or semantic meaning to the compiler. These are called punctuators. **Examples** are: # ; ' " ( ) [ ] { }

e)Operators: An operator is a symbol that tells the compiler about a specific operation. T C++ provides different types of operators like arithmetic, relational, logical, assignment, conditional, etc.

Chapter 6 : Data types and Operators.

Fundamental data types: Fundamental data types are defined in C++ compiler. They are also known as built-in data types. .The five fundamental data types in C++ are char , int , float , double and void .

a)int data type: to store integer numbers, occupies 4 bytes in memory

b) char data type: to store characters, occupies 1 bytes in memory

c) float data type: for storing fractional numbers, occupies 4 bytes in memory

d) double data type: for storing double precision floating point numbers. It takes 8 bytes of memory

e) void data type: specifies an empty set of data, occupies no memory (0 bytes).

Variables: Variables are the names given to memory locations. There are three important aspects for a variable.

i. Variable name: It is a symbolic name (identifier) given to the memory location through which the content of the location is referred to.

ii. Memory address: All the variables are connected to one or more memory locations in RAM. The base address of a variable is the starting address of the allocated memory space. The address is also called the L-value of a variable.

iii. Content of a Variable: The value stored in the location is called content of the variable. This is also called the R-value of the variable.

Operators: An operator is a symbol used to perform an operation.

i) Operator Classification based on number of operands

Unary operator: Unary operator functions on a single operand eg: a++, -b, d—

Binary operator: Binary operator acts on two operands. eg a+b, c/q

Ternary operator: A Ternary operator (?:) acts on three operands.

ii) Operator Classification based on its Use

Arithmetic operators: Arithmetic operator are used for arithmetic operations (+,-,*,/,%)

Relational operators: Relational operators are used to compare values.(<,>,<=,>=)

Logical operator: Logical operators allows us to combine two or more conditions.(&&,||,!)

Input operators: The >> operator is called extraction or get from operator. Used to store data from keyboard in a memory location. Eg: cin>>num;

Output operators: The operator << is called insertion or put to operator. Used for output operations
eg: cout<<num;

Cascading of I/O operators: The multiple use of input or output operators in a single statement is called cascading of I/O operators. Eg: cin>>a>>b>>c;

```
cout<<a<<b<<c;
```

Assignment operator (=): to store a value in a variable assignment operator(=) is used. Eg: num=50;

=	==
Assignment Operator	Equal to Operator
Used to assign a value to a variable	Used for Equality checking
As result value is stored in variables memory location	The result will be either true or false
eg: a=10;	eg: a==10

Type conversion: converting one data type to another data type. There are two types of type conversion

i)Implicit type conversion (Type promotion) : is performed by C++ compiler internally. C++ converts the lower sized operands to the data type of highest sized operand. eg: float x; x= 5/2;

ii) Explicit type conversion(Type casting): Programmer explicitly casts to the desired type. eg: x=(float)p/q;

Statements: are the smallest executable unit of a programming language.

i)Declaration Statements: to tell the compiler about the type of data that will be stored in it. Eg: int admno; float avg,per;

ii)Assignment Statement: to assign a value to a variable using assignment operator eg: a=50;

iii)Input Statement: Allow user to store data in memory eg: cin>>num;

iV)Output Statement: Makes results available to users through any output devices. Eg: cout<<num;

Structure of a C++ program

```
#include<header file>
using namespace identifier;
int main()
{
Statements;
        ;
        ;
return 0;
}
```

```
#include<iostream>
using namespace std;
int main()
{
cout<<"My Program";

return 0;
}
```

Pre-processor directive: Pre-processor directives are compiler directive which instructs the compiler to process information before actual compilation. Pre-processor directives begins with # (hash) symbol. Eg:-#define, #undef, #include..

Concept of namespace: Different identifiers are associated to a particular namespace. It is a group name in which each item is unique in its name.

Header files: Header files contain the information about functions, objects and predefined derived data types that are available along with compiler eg: #include<iostream>

The main() function: The execution of a C++ program starts and ends within main().

Chapter 7. Control Statements.

Statements used for altering the default flow of execution.

(i) Decision making / selection statements: statements for the selected execution based on a condition.

if statement: used to select a set of statements for execution based on a condition

Syntax:

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

eg:

```
if (score >= 18)
cout << "You have Passed";
```

if...else statement: used to select a set of statements for execution when a condition is true and some actions when the condition becomes False.

Syntax:

```
if (test expression)
{
statement block 1;
}
else
{
statement block 2;
}
```

eg:

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

Switch Statement:

Eg:

```
switch (day)
{
case 1: cout << "Sunday";break;
case 2: cout << "Monday";break;
case 3: cout << "Tuesday";break;
default: cout << "Wrong input";
}
```

switch statement	else if ladder
Permits multiple branching.	Permits multiple branching.
Evaluates conditions with equality operator only.	Evaluate any relational or logical expression.
Case constant must be an integer or a character type value.	Condition may include range of values and floating point constants.
When no match is found, default statement is executed.	When no expression evaluates to True,else block is executed.
break statement is required for exit from the switch statement.	Program control automatically goes out after the completion of a block.
More efficient when the same variable or expression is compared against a set of values for equality.	More flexible and versatile compared to switch.

else if ladder : It is used in programs when multiple branching is required. Different conditions will be given and each condition will decide which statement is to be selected for execution. eg.

```
if (score >= 80)
        cout<<"A Grade";
                else if (score >= 60)
                        cout<<"B Grade ";
                                else if (score >= 40)
                                        cout<<"C grade";
```

```
else if (score >= 30)
        cout<<"D grade";
            else
        cout<<"E Grade";
```

ii)Iteration statements or Looping statements: to perform the repeated execution of one or more statements. A variable like the counter will be used to construct a loop. This variable is generally known as ***loop control variable*** because it actually controls the execution of the loop

Four elements of a loop:

Initialisation: Before entering a loop, its control variable must be initialised. The initialization statement is executed only once, at the beginning of the loop.

Test expression: It is a relational or logical expression whose value is either True or False.

Update statement: The update statement modifies the loop control variable by changing its value.

Body of the loop: The statements that need to be executed repeatedly constitute the body of the loop. It may be a simple statement or a compound statement

`while statement:`

Syntax:

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

Eg:

```
int n = 1;
while(n <= 10)
{
cout << n << " ";
++n;
}
```

`for statement:`

Syntax:

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

Eg:

```
for (n=1; n<=10; ++n)
cout << n << " ";
```

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

`do...while statement:`

Syntax:

```
initialisation of loop control variable;
do
{
body of the loop;
Updation of loop control variable;
} while(test expression);
```

Eg:

```
do
{
cout << k << '\t';
++k;
} while(k<=3);
```

Jump Statements: The statements that facilitate the transfer of program control from one place to another.They are return, goto, break and continue statements.

break statement: When a break statement is encountered in a loop (for, while, do-while), it takes the program control outside the immediate enclosing loop.

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

```
for (i=1; i<=10; ++i)
{
if (i==6)
continue;
cout<<i<<"\t";
}
```

This code gives the following output: 1 2 3 4 5 7 8 9 10

break statement	continue statement
• Used with `switch` and loops. • Brings the program control outside the `switch` or loop by skipping the rest of the statements within the block. • Program control goes out of the loop even though the test expression is True.	• Used only with loops. • Brings the program control to the beginning of the loop by skipping the rest of the statements within the block. • Program control goes out of the loop only when the test expression becomes False.

Table 7.4: Comparison between the `break` *and* `continue` *statements*

Chapter 8 - Arrays

Array: An array is a collection of elements of the same type placed in contiguous memory locations.The position is called the index or subscript value. In C++, the array index starts with zero.

Declaring arrays: The syntax for declaring an array in C++ is as follows.

data_type array_name[size];

example: int num[10];

Array initialisation: Array elements can be initialised in their declaration statements in the same manner as in the case of variables, except that the values must be included in braces, as shown in the following examples:

```
int score[5] = {98, 87, 92, 79, 85};
char code[6] = {'s', 'a', 'm', 'p', 'l', 'e'};
float wgpa[7] = {9.60, 6.43, 8.50, 8.65, 5.89, 7.56, 8.22};
```

Accessing elements of arrays: Array is accessed element-wise That is, only one element can be accessed at a time. The element is specified by the array name with the subscript.

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

1.Traversal: Accessing each element of an array at least once to perform any operation is known as traversal operation.

2.Sorting: Sorting is the process of arranging the elements of the array in some logical order.

a. Selection sort: To sort an array in ascending order, the selection sort algorithm starts by finding the minimum value in the array and moving it to the first position. At the same time, the element at the first position is shifted to the position of the smallest element.

b.Bubble sort: Bubble sort is a sorting algorithm that works by repeatedly stepping through lists that need to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order.

3.Searching: Searching is the process of finding the location of the given element in the array.

a.Linear search: Linear search or sequential search is a method for finding a particular value in a list.

b.Binary search : Binary search is an algorithm which uses minimum number of searches for locating the position of an element in a sorted list, by checking the middle, eliminating half of the list from consideration, and then performing the search on the remaining half.

Linear search method	Binary search method
• The elements need not be in any order	• The elements should be in sorted order
• Takes more time for the process	• Takes very less time for the process
• May need to visit all the elements	• All the elements are never visited
• Suitable when the array is small	• Suitable when the array is large

Table 8.1: Comparison of linear and binary search methods

Program 2.2: Traversal of an array

```
#include <iostream>
using namespace std;
int main()
{
    int a[5], i;
    cout<<"Enter the elements of the array :";
    for(i=0; i<5; i++)
        cin >> a[i];  //Reading the elements
    for(i=0; i<5; i++)
        a[i] = a[i] + 1; // A case of traversal
    cout<<"\nNow value of elements in the array are...\n";
    for(i=0; i<5; i++)
        cout<< a[i]<< "\t";  // Another case of traversal
    return 0;
}
```

Chapter 9 : String handling and I/O Functions

Character array declaration: char my_name[10];

initialization method

char my_name[10]={'N', 'i', 'k', 'e', 't', 'h'};

Input/Output operations on strings:

cin	gets()
cin does not accept the input of spaces	gets() accepts continuous input, including spaces, TAB
Used to enter numbers.	Receive a string, and the space, TAB
Header file iostream	Header file cstdio

Stream functions for I/O operations: These functions are generally called stream functions since they allow a stream of bytes (data) to flow between memory and objects.

A. Input functions: These functions allow the input of character and string data.

i. get() :It can accept a single character or multiple characters (string) through the keyboard.

Example:

```
char ch, str[10];
ch=cin.get(ch);      //accepts a character and stores in ch
cin.get(ch);         //equivalent to the above statement
cin.get(str,10);     //accepts a string of maximum 10 characters
```

ii. Getline(): It accepts a string through the keyboard. The delimiter will be Enter key, the number of characters or a specified character.

Examples:

```
char ch, str[10];
int len;
cin.getline(str,len);          // With 2 arguments
cin.getline(str,len,ch);       // With 3 arguments
```

B. Output functions: Output functions like put() and write() allow a stream of bytes to flow from memory into an output object.

i. put(): It is used to display a character constant or the content of a character variable given as argument.

```
char ch='c';
cout.put(ch);          //character 'c' is displayed
cout.put('B');         //character 'B' is printed
cout.put(65);          //character 'A' is printed
```

ii. write() : This function displays the string contained in the argument. example given below.

```
char str[10]="hello";
cout.write(str,10);
```

The above code segment will dispaly the string hello followed by 5 white spaces, since the second argument is 10 and the number of characters in the string is 5

Chapter 10 : Functions

Concept of modular programming: In programming, the entire problem will be divided into small sub problems that can be solved by writing separate programs. This kind of approach is known as modular programming.

Merits of modular programming:

Reduces the size of the program: In some cases, certain instructions in a program may be repeated at different points of the program.The modular approach helps to isolate the repeating task and write instructions for this.

Less chances of error: When the size of the program is reduced, naturally syntax errors will be less in number. The chances of logical error will also be minimized.

Reduces programming complexity: Modularization reduces the programming complexity by bringing down our mind to a simplified task at a time, reducing the program size and making the debugging process easy.

Improves reusability: A function written once may be used later in many other programs, instead of starting from scratch. This reduces the time taken for program development.

Demerits of modular programming: Proper breaking down of the problem is a challenging task. Care should be taken while setting the hierarchy of the execution of the modules.

String Functions: for the manipulation of strings,the header file **cstring**

Functions	Syntax / Example	Operation
strlen()	strlen(string)	To find the length of a string
strcpy()	strcpy(string1, string2)	To copy one string into another
strcmp()	strcmp(string1, string2)	To compare two strings
strcmpi()	strcmpi(string1, string2)	To compare two strings ignoring cases

Mathematical functions: for mathematical calculations, the header file **cmath**

Functions	Syntax / Example	Operation
abs()	abs(int)	To find the absolute value of an integer.
sqrt()	sqrt(double)	To find the square root of a number.
pow()	pow(double, int)	To find the power of a number.
sin()	sin(double)	To finds the sine value of an angle.
cos()	cos(double)	To find the cosine value of an angle.

Character functions: used to perform various operations of characters, header file **cctype**

Character functions: header file **cctype**

Functions	Syntax / Example	Operation
isupper()	isupper(char)	To check whether a character is in upper case or not.
islower()	islower(char)	To check whether a character is in lower case or not.
isalpha()	isalpha(char)	To check whether a character is alphabet or not
isdigit()	isdigit(char)	To check whether a character is digit or not.
isalnum()	isalnum(char)	To check whether a character is alphanumeric or not.
toupper()	toupper(char c)	This function is used to convert the given character into its uppercase.
tolower()	tolower(char c)	This function is used to convert the given character into its lowercase.

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 , which is optional, the body comprises C++ statements required to perform the task assigned to the function.

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 constants, variables or expressions used in the function call are known as actual (original) arguments.

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

Chapter 11 Computer Networks

Computer network is a group of computers and other computing hardware devices connected together.

Advantages of computer Networks

Resource sharing: The sharing of available hardware and software resources in a computer network is called resource sharing

Price-performance ratio: Since resources can be shared the cost purchasing licensed software for each computer can be reduced by purchasing networked version of such resources

Communication: Computer network helps users to communicate with any other uses of that network through its services like-mail, chatting, video conferencing etc

Reliability: it is possible to replicate or backup data/ information in multiple computers using the network. This helps user to retrieve date in the case of failures in accessing of data.

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

Some key terms

Bandwidth : bandwidth measures the amount of data that can be send over a specific connection in a given amount of time.

Noise: Noise 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 devices:

Switch : Like hub Switch is also used to connect computers on a network. But it an intelligent device and transmits the received data to the destination only. A switch performs better than in a bust network, since it generate less network traffic. Expensive than hub.

Repeater : Repeater is a device that regenerates incoming, electrical, wireless or optical devices through communication medium.

Bridge : A bridge is a device used to segmentise a network. When a data packet reaches the bridge it find out the receiver if it is addressed to node at the other side, will be allowed to pass the bridge.

Router : A device that can interconnect two networks of the same type using the same protocol. It is more intelligent than bridge. It can find the optimal path for data packets to travel and reduce the amount of traffic in the network.

Gateway : A gateway is a device that can interconnect two different networks having different protocol. It can translate one protocol to another. It can also find the optimal path.

Data Terminal Equipments: Data terminal equipment is a device the flow to or from a computer.

Modem : A modem is a device used to for communication between computer through telephone lines

Network topologies

Topology : 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 terminator is attached each end of the bus. If a node has to send data to another node, it sends to the bus. The signal travels through the bus and each node check the address and intended node will accept the data. When the signal reaches the end of the bus terminator removes the data from the bus.

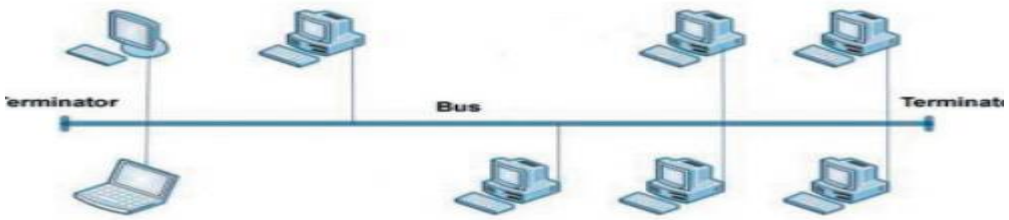


Fig. 8.23 : Bus topology

Star topology: In star topology each node is connected a hub/ switch. If a node has to send some information to any other node, it sends the signal to the switch or hub. Signal then broadcasted to all other nodes in the case of hub and forwarded to intended node in the case of switch

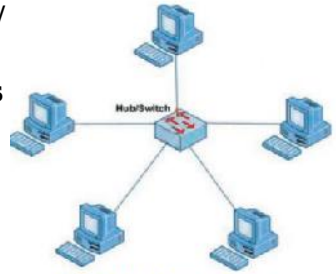


Fig. 8.24 : Star topology

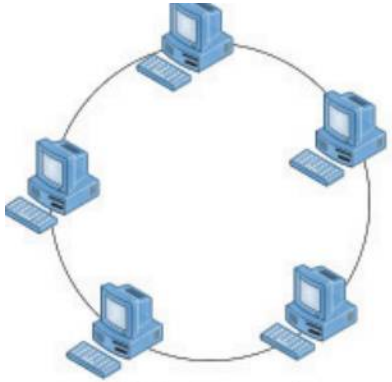


Fig. 8.25 : Ring topology

Ring topology: in ring topology all nodes are connected using a cable that loops the ring or circle. A ring topology forms a ring.Data travels only in one direction in a ring. Each node regenerates the signal and pass to next node until it reaches the intended node reaches.

Mesh topology : In mesh topology, every node is connected to other nodes, So there will be more than one path between two nodes. Network will not fail even if one path between the nodes fails.

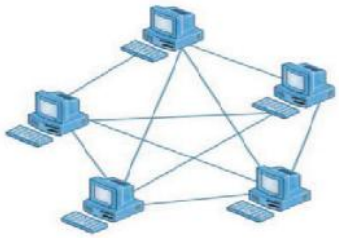


Fig. 8.26 : Meshopology

Identification of Computers on a network

Media Access Control (MAC) address: **A universally unique address (12 digit hexadecimal number) assigned to each NIC (Network Interface Card) by its manufacturer. First Six digit for the ID number of the manufacturer and remaining serial number of adapter.**
MM : MM : MM : SS : SS : SS or MM – MM – MM – SS – SS – SS
eg. 00:A0:C9 : 14:C8:35

IP address: **An IP address is a group of four bytes (or 32 bits) each of which can be a number from 0 to 255. IP address is provided to each machine by the network administrator or by the Internet service provider.**
eg.168 . 20 . 1 . 2

Chapter 12 Internet and Mobile Computing

Services on Internet

www **Search engines** **E-mail** **Social media**

WWW:**WWW is a client- server system, which consists of millions of clients and servers connected together accessed using a URL(Uniform Resource Locator).**

Browser: **It is software that helps to view a webpage. A browser can display text, images, videos, hyperlinks etc of a webpage. Eg Google Chrome, Internet Explorer, Mozila Firefox, Opera, Safari etc**

Web Browsing: **Web Browsing is the process of visiting the web site of various companies, organizations for any kind of information.**

Search Engines: Search engine programs search documents available on World Wide Web for specified keywords and return a list of the documents/web pages matching the keywords. Search engine web sites use programs called web crawlers or spiders or robots to search the web. Eg:. For search engine are Google, Yahoo Search, Bing, Ask etc

Email (Electronic Mail): It is an electronic message send from one computer to another through the internet.

Cyber security:

Computer Virus:

It is computer program that attachés itself to another program or file to spread from one computer to another and can cause damages to your computer, delete files etc.

Trojan horse: **Trojan Horse appear to be useful s/w but will actually do damage once you installed on the computer. A Trojan horse can delete files, destroy information on the system.**

Hacking: **It is the process of gain unauthorized access to data in a system or computer. The person who involved in hacking is called Hacker.**

Phishing: Phishing is an attempt to acquire information such as usernames, passwords, and credit card details of a person by using a fake website in the place of original website of a bank/financial organization.