Chapter 1: The Discipline of Computing

Abacus: Considered as the first computer for basic arithmetical calculations.

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

Analytical Engine: Developed by Charles Babbage, the 'Father of Computer'. The Engine had a 'Store' (memory) and a separate 'Mill' (processor). *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

Von Neumann architecture:

The mathematician John Von Neumann designed computer architecture. It consists of a central processing unit (CPU) containing arithmetic logic unit (ALU) and control unit (CU), input-output unit and a memory for storing data and instructions.
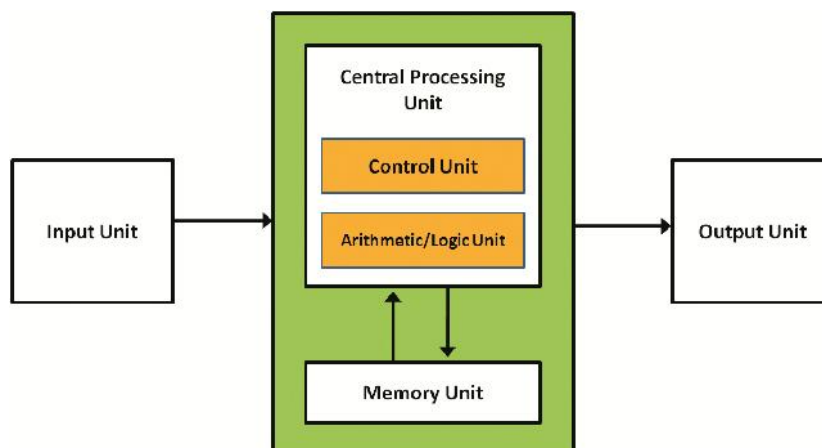
Chapter 2: Data Representation and Boolean Algebra

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

Number Conversion procedure:

| Conversion	Procedure
Decimal to Binary	Repeated division by 2 and grouping the remainders
Decimal to Octal	Repeated division by 8 and grouping the remainders
Decimal to Hexadecimal	Repeated division by 16 and grouping the remainders
Binary to Decimal	Multiply binary digit by place value(power of 2) and find their sum
Octal to Decimal	Multiply octal digit by place value (power of 8) and find their sum
Hexadecimal to Decimal	Multiply hexadecimal digit by place value (power of 16) and find their sum
Octal to Binary	Converting each octal digit to its 3 bit binary equivalent
Hexadecimal to Binary	Converting each hexadecimal digit to its 4 bit binary equivalent
Binary to Octal	Grouping binary digits to group of 3 bits from right to left
Binary to Hexadecimal	Grouping binary digits to group of 4 bits from right to left
Octal to Hexadecimal	Convert octal to binary and then binary to hexadecimal
Hexadecimal to Octal	Convert hexadecimal to binary and then binary to octal

Number representation methods: (i) Sign and magnitude representation (ii) 1's complement representation (iii) 2's complement representation.

Character Representations

ASCII: American Standard Code for Information Interchange. It uses 7 bits to represent a character. Another version ASCII-8 uses 8 bits.

Unicode: Originally used 16 bits which can represent up to 65,536 characters. Unicode can represent characters in almost all written languages of the world.
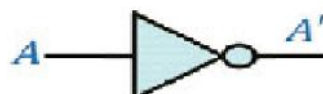
Boolean operations: The operations performed on the Boolean values 0s and 1s. The operations are OR (Logical Addition), AND (Logical Multiplication), NOT (Logical Negation).
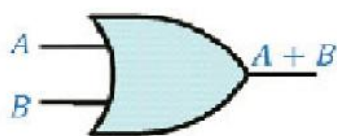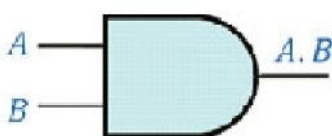
Logical OR gate   Logical AND gate   Logical NOT gate

Chapter 3: Components of Computer System

Processor: CPU is referred to as the processor. It is responsible for all computing and decision making operations. It coordinates the working of a computer. It has ALU, CU and registers as components.

Important registers inside a CPU:
Accumulator, Memory Address Register (MAR), Memory Buffer Register (MBR), Instruction Register (IR), Program Counter (PC).

Ports: Used to connect external devices. VGA, PS/2, USB, HDMI, Ethernet are examples.

RAM: Random Access Memory, Volatile, CPU can directly access it.

Measuring units of memory:

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: Electronic waste may be defined as discarded computers, office electronic equipment, entertainment devices, mobile phones, television sets and refrigerators.

e-Waste disposal methods: Re-use, incineration (combustion process in which the waste is burned in incinerators at a high temperature), recycling, (the process of making new products from a product that has originally served its purpose) and land filling.

System software: The components of system software are Operating system, Language processors and Utility software.

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

Function of OS: Process management, memory management, file management, device management, security management and command interpretation.

Language processors: These are the system programs that translate programs written in high level language into machine language. Eg: Interpreter (converts a HLL program into machine language line by line) and Compiler (translates a program written in high level language into machine language).

Free and Open source software: Gives the user the freedom to use, copy, distribute, examine, change, and improve the software. GNU/Linux is a free and open source operating system. Firefox, Libre Office are also examples.

*Four freedoms for free and open source software:*

Freedom 0, Freedom 1, Freedom 2, Freedom 3

Chapter 4 – Principles of Programming and Problem Solving

Stages of problem solving: (i) Problem identification (ii) Preparing algorithms and flowcharts (iii) Coding the program using programming language (iv) Translation (v) Debugging (vi) Execution and Testing and (vii) Documentation

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

Flowchart: The pictorial representation of an algorithm.

Flowchart Symbols: Start/Stop   Input / Output   Process   Decision

Flow lines   Connectors
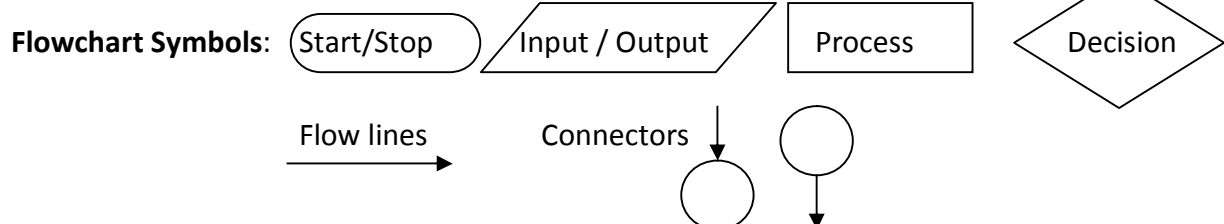
Debugging: The process of detecting and correcting the errors in a program. The following types of errors are corrected.

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

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

Run time error: These errors occur unexpectedly when computer becomes unable to process some improper data.

Chapter 5: Introduction to C++ Programming

Tokens: The fundamental building blocks of the program. Five types of tokens – Keywords, Identifiers, Literals, Punctuators and Operators.

Keywords: The words (tokens) that convey a specific meaning to the language compiler.

Identifiers: These are the user-defined words. Variables, labels, function names are identifiers.

Rules for naming identifiers: The first character must be a letter or underscore (_). White space and special characters are not allowed. Keywords cannot be used. Eg: Num, a2b, _var

Literals: Four types – Integer literals, Floating point literals, Character literals and String literals.

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

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

Operator: A symbol that tells the compiler about a specific operation.

Chapter 6: Data Types and Operators

Fundamental data types: Also known as basic data types. Five fundamental data types - char, int, float, double and void.

Variable: Identifier of memory location. Syntax for declaration: data_type variable;

Eg: `int a;   float b;`   Example for variable initialisation: `int n=10;`

Variable has two values: l- value (address), r-value (content).

Operators: The tokens or symbols that trigger computer to carry out operations. Classified into three –unary, binary and ternary.

Arithmetic operator (+, -, *, /, %), Relational operator (<, <=, >, >=, ==, !=), Logical operator (&&, ||, !), Assignment operator (=), Increment operator (++), Decrement operator (--), Conditional operator (? :).

Modulus operator (%): Also called as mod operator. It gives the remainder value during arithmetic division.

Expressions: Combination of operators and operands. Each expression returns a value. Three types of expressions – arithmetic, relational and logical.

Statements: Declaration statement, input statement, output statement.

Cascading of I/O operators: Multiple use of input or output operators in a single statement. Eg: `cin>>a>>b;      cout<<a<<b;`

Type conversion: Conversion of the data type of an operand into another type. Done in two ways: implicitly and explicitly.

Joy John's CS Capsule

Type promotion: It is the implicit type conversion is performed by C++ compiler internally.

Type casting: It is the explicit type conversion and is done by the programmer.

main() function: An essential function in every C++ program. The execution starts at main() and ends within main().

Structure of C++ program:

#include<iostream.h>	→ Pre processor directive to include header file
using namespace <std>	→ To use cin and cout independently
int main()	→ Essential function in C++ program
{	
statements;	→ Program statements
return 0;	→ To end the program
}	

Chapter 7: Control Statements

<table>
<tr><td valign="top">

`if statement`

Syntax:
```
if (test expression)
 {
     statement block;
 }
```
Here the `test expression` represents a condition. If the `test expression` is, a statement or a block of statements associated with `if` is executed.

</td><td valign="top">

`if − else statement`

Syntax:
```
if (test expression)
{
     statement block 1;
}
else
{
     statement block 2;
}
```
If the `test expression` evaluates to True, only the `statement block 1` is executed. If the `test expression` evaluates to False `statement block 2` is executed.

</td></tr>
</table>

When we write an `if` statement inside another `if` block, it is called **nesting**.

`else if` ladder

Syntax:
```
if (test expression1)
    statement block 1;
else if (test expression2)
    statement block 2;
....................
else
    statement N;
```

`switch statement`

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

Joy John's CS Capsule

```
                              :
    case constant_n-1   : statement block n-1;
                          break;
    default             : statement block n;
}
```

Looping (Iteration) Statements

Four elements of a loop: Initialisation, Test expression, Update statement, Body of the loop

`while` statement

It is an entry-controlled loop. The condition is checked first and if it is True the body of the loop will be executed. The syntax of `while` loop is:

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

`for` statement

It is also an entry-controlled loop in C++. All the three loop elements (initialisation, test expression and update statement) are placed together in `for` statement. The syntax is:

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

do...while statement

In the case of `for` loop and `while` loop, the test expression is evaluated before executing the body of the loop. Its syntax is :

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

`break` V/s `continue`

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

Joy John's CS Capsule

Chapter 8: Arrays

Arrays are used to store a set of values of the same type under a single variable name.

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

```
data_type array_name[size];
```

Eg: `int num[10];`

This statement declares an array named `num` that can store 10 integer numbers.

Array elements can be initialised in their declaration as shown in the following example:

`int score[5] = {98, 87, 92, 79, 85};`

Accessing array elements:

Uses the subscript within brackets along with array name. Eg: socre[1] accesses the 2nd element of the array score.

To access all elements of the above array use the statement:

```
for (i=0; i<5; i++)
    cout<<score[i];
```

Operation on arrays:

Traversal: Visiting all elements of an array. (The above code is an example)

Searching: Checking the presence of an element in the array.

Sorting: Arranging the elements in ascending or descending order.

Chapter 9: String Handling and I/O Functions

Character arrays are declared to store strings.

`char name[20];` is an array that can store a string having a maximum of 19 characters.

`char name[]= "Hello";` initialises the array name[] with the delimiting character '\0' at 5th subscript of the array.

I/O Functions for strings:

Type of data	Stream functions (iostream.h header file)	Operation
Character functions	cin.get()	Character input
	cout.put()	Character output
String functions	cin.getline(), cin.get()	String input
	cout.write()	String output

Chapter 10: Functions

Modular programming: Dividing the entire problem into small sub tasks which are solved by writing separate modules (sub programs).

Merits:

- Reduces the size of the program.
- Less chance of error occurrence.
- Improves reusability.

Type	Functions	Syntax / Example	Operation
String Functions (string.h)	**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.
Mathematical Functions	**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.
Character Functions (ctype.h)	**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;
    return val;
}
```

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, is a list of parameters, i.e. a list of variables preceded by data types and separated by commas. The `body` comprises of C++ statements required to perform the task assigned to the function. Function usually returns a value and it should be of the data type of the function.

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**.

Chapter 11: Computer Networks

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

Advantages: Resource sharing, Price-performance ratio, Communication, Reliability, Scalability.

Bandwidth: The amount of data sent over a specific connection in a given amount of time. It is measured in bits per second (bps).

Noise: Unwanted electrical or electromagnetic energy that lowers the quality of data signal.

Node: Any device (computer, scanner, printer, etc.) which is directly connected to a computer network.

Communication devices: NIC, Hub, Switch, Repeater, Bridge, Router, Gateway.

**Switch**: An intelligent device that connects several computers to form a network.

Bridge: A device used to split a network into different segments and interconnected.

Router: A device that can interconnect two networks of the same type using the same protocol. It is more intelligent than bridge.

Gateway: A device that interconnects two different networks having different protocols.

Modem: It converts digital signals to analog signals and converts the analog signals back to digital signals.

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

Bus Topology   *Star Topology*   *Ring Topology*
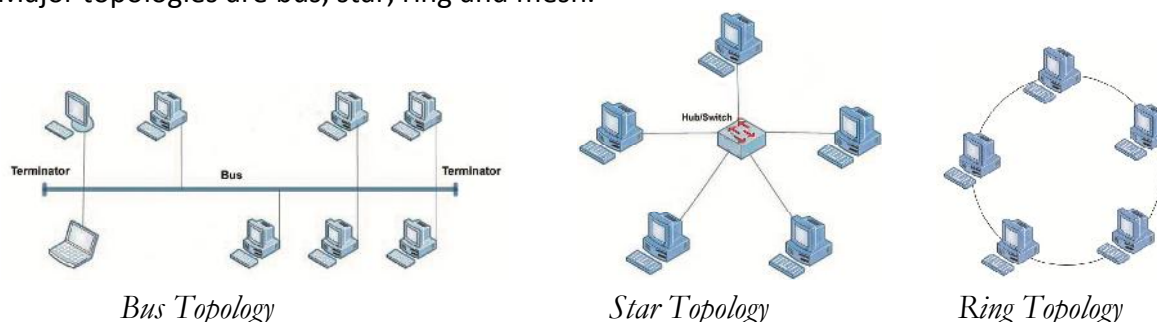
IP Address: An IP address is a unique 4 part numeric address assigned to each node on a network for their unique identification. Each part is a number from 0 to 255.
Eg: 148.78.250.12

MAC Address: A Media Access Control (MAC) address is a universally unique address (12 digit hexadecimal number) assigned to each NIC (Network Interface Card) by its manufacturer.

Chapter 12 – Internet and Mobile Computing

Services on the Internet: Services like WWW, e-mail, search engines, social media, etc. are widely used throughout the globe.

Web browser: It is a software use to retrieve or present information and to navigate through web pages in the World Wide Web. Eg: Google Chrome, Internet Explorer, Mozilla Firefox, Opera, and Safari.

Search engines: Internet search engine websites are special programs that are designed to help people to find information available in World Wide Web. Search engine web sites use programs called web crawlers or spiders or robots to search the web.

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

Sections of e-mail: To (Recipient Address), Cc (Carbon copy to the secondary recipients), Bcc (Blind carbon copy to the tertiary recipients), Subject, Content, Attachments.

Advantages of e-mail: Speed, easy to use, provision of attachments, environment friendly, easy reply to an e-mail, cost-effective, available anywhere anytime.

Types of social media: Internet forums, social blogs, micro blogs, wikis, social networks, content communities and a lot more.

Internet forums: It is an online discussion website where people can engage in conversations in the form of posted messages. Eg: Ubuntu Forum

Social blogs: It is a discussion or informational website consisting of entries or posts displayed in the reverse chronological order i.e., the most recent post appears first. Eg: Blogger.com, Wordpress.com.

Microblogs: They allow users to exchange short sentences, individual images or video links. It offers a communication mode that is spontaneous and can influence public opinion. Twitter.com is a popular micro blogging site.

Wikis: Wikis allow people to add content or edit existing information in a web page, to form a community document. Wiki is a type of content management system. Eg: wikipedia.org.

Social networks: These sites allow people to build personal web pages and then connect with friends to communicate and share content. We can share text, pictures, videos, etc. and comment to the posts. Eg: facebook.com, linkedin.com.

Content communities: These are websites that organise and share contents like photos, videos, etc. Eg: Youtube.com, flickr.com.

Advantages of social media: Bring people together, help to plan and organise events, promoting business, enhance social skills.

Limitation of social media: Intrusion to privacy, addiction, spread rumours.

Computer virus: It is a program that attaches itself to another program or file enabling it to spread from one computer to another without our knowledge and interferes with the normal operation of a computer.

Trojan horse: It will appear to be a useful software but will actually do damage once installed or run on your computer.

Hacking: It is a technical effort to manipulate the normal behaviour of network connections and connected systems. Computer experts perform hacking to test the security and find the vulnerabilities in computer networks and computer systems. Such computer experts are often called '**white hats**' and such hacking is called ethical hacking. Computer criminals break into secure networks to destroy data or make the network unusable for those who are authorised to use the network. Such criminals are called '**black hats**'. There are '**grey hat** hackers', who sometimes act illegally, though with good intentions, to identify the vulnerabilities.

Phishing: It is an attempt to acquire information such as usernames, passwords and credit card details by posing as the original website, mostly that of banks and other financial institutions. Phishing websites have URLs and home pages similar to their original ones. The act of creating such a misleading website is called spoofing.