Focus Area Notes

Computer Science

T Subair
GVHSS Payyoli
Kozhikode
Mob: 9946435670

Computer Science
Focus Area

1. The Discipline of Computing	Evolution of Computing machines (Abacus, Difference engine, Analytical engine), Generations of computers
2. Data Representation and Boolean Algebra	Number systems, Number conversions Decimal to nondecimal and reverse, Shortcut methods (avoid fractional conversion) Representation of integers (Sign & Magnitude, 1's and 2's compliments) and characters (ASCII & Unicode), Boolean operators (AND, OR, NOT) and logic gates, Simple circuit designing.
3. Components of the Computer System	Processor, Ports, Memory (RAM only with measuring units), e-Waste and disposal methods, System software (OS, Language processors compiler and interpreter), Free and open source software.
4. Principles of Programming and Problem Solving	Phases in programming (Listing only), Debugging (Types of errors), Flowchart symbols, Development of algorithms and flowcharts to solve simple problems only (except looping).
5. Introduction to C++ Programming	Tokens and classification with examples
6. Data types and Operators	Fundamental data types, Variables, Operators and classifications, Type conversion, Various types of statements, Structure of C++ program.
7. Control Statements	Decision making statements (if, if-else, if-else if, switch), Iteration statements (while, for, do-while) - syntax and working, (Nesting not required), Jump statements (break, continue). (No programming)

8. Arrays	Declaration, Initialisation, Accessing elements, Operations (listing only with concept), Traversal operation with simple program.
9. String Handling and I/O Functions	Array declaration for string and initialisation, Input/Output operations, Use of get(), getline(), put(), write() functions. (No programming)
10. Functions	Modular programming and merits, Predefined functions (string, mathematical, character), User-defined functions (Syntax, Concept of arguments and return value). (No programming)
11. Computer Networks	Advantages of network, Key terms (Bandwidth, noise, node), Communication devices (switch, router, gateway, bridge, modem), Network topologies, Identification of computers over network (MAC, IP)
12. Internet and Mobile Computing	Services on Internet (Working procedure is not required), Cyber security (Computer virus, Trojan horse, hacking, phishing).

Chapter 1

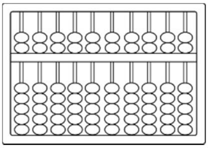
The Discipline of Computing

Focus Area: Evolution of Computing machines (Abacus, Difference engine, Analytical engine), Generations of computers.

Computer Science is the study of computers and computational systems. It deals mostly with software and software systems including their theory, design, development, and application.

Evolution of the Computing Machine

1. **Abacus**



The first computing machine Abacus was discovered by Mesopotamians around 3000BC. 'Abacus' means calculating board. Abacus is an instrument for performing calculations by sliding counters along rods. It works on the basis of place value system.

2. **Difference Engine**

The difference engine is a mechanical calculator first developed by Charles Babbage in 1822. It is capable of computing several sets of numbers and making a hard copies of the results.

3. **Analytical Engine**

Charles Babbage developed Analytical engine in 1833. It has many essential features found in the modern digital computer. It was programmable using punched cards, an idea borrowed from the Jacquard loom. The Engine had a 'Store' (memory) where numbers and intermediate results could be held, and a separate 'Mill' (processor) where the arithmetic processing was performed.

Charles Babbage is known as the 'Father of Computer' due to his inventions, the difference engine and the analytical engine.

Generations of computers

Criteria	Generation				
	First	**Second**	**Third**	**Fourth**	**Fifth**
Technology	Vacuum Tube	Transistor	IC	Microprocessor	Artificial Intelligence
Operating System	None	None	Yes	Yes	Yes
Language	Machine	Assembly	High level	High level	High level
Period	1940-56	1956-64	1964-71	1971-Present	Present and Future

1. **First generation computers (1940-56)**
   Vacuum tubes were used in first generation computers. Stored program concept is used in this generation. Punched cards and paper tapes are used for input and output was printouts. The first general purpose computer ENIAC (Electronic Numerical Integrator and Calculator), EDVAC and commercial computer UNIVAC (Universal Automatic Computer) were in this generation. 1G computers were bulky.
2. **Second generation computers (1956-64)**
   Transistors were used in second generation computers. Computers became smaller, powerful faster and less expensive in 2G. Programming language concept was developed.
3. **Third generation computers (1964-1971)**
   IC (Integrated Circuit) chips were used in third generation computers. Speed and efficiency increased and the size and cost are reduced. Keyboards and monitors were introduced. High level languages like BASIC were developed.
4. **Fourth generation computers (1971-Present)**

Microprocessors are used in fourth generation computers and we use these computers today. These computers are even smaller and faster. Computer network and internet came in to existence.

5. **Fifth generation computers (Present and future)**
   Fifth generation computers are based on Artificial Intelligence (AI). AI tries to simulate human intelligence. Speech recognition, face recognition, robotic vision etc. are some of the milestones achieved with AI in modern computers.

EXCERCISES

1 Score Questions

1. …. is known as the father of Computer.
2. What is an Analytical engine?
3. Transistors were introduced in ….. generation computers.

2 Score Questions

1. Write a short note about difference engine.
2. What is an abacus?

3 Score Questions

1. Explain different generations of computers

Chapter 2

Data Representation and Boolean Algebra

Focus Area: Number systems, Number conversions Decimal to non-decimal and reverse, Shortcut methods (avoid fractional conversion) Representation of integers (Sign & Magnitude, 1's and 2's compliments) and characters (ASCII & Unicode), Boolean operators (AND, OR, NOT) and logic gates, Simple circuit designing.

Data representation is the method used to represent data in a computer. A computer can process only binary data.

Number System

There are four types of number systems namely **Decimal, Binary, Octal and Hexadecimal.** The number of symbols (digits) used in a number system is called the **base** (radix) of the number system. We normally use decimal number system in our daily life. Computer uses Binary for processing. Each digit of a binary number is called a **bit** (binary digit). Octal and Hexadecimal systems are used to save memory and easier conversion to and from binary.

The table below shows the base and the digits used in each number system

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

Note 1: -

1. MSB (Most Significant Bit) - It is the left most bit in a binary number.

2. LSB (Least Significant Bit) - It is the right most bit in a binary number.
3. MSD (Most Significant Digit) – It is the left most digit in a number system.
4. LSD (Least Significant Digit) – It is the right most digit in a number system.

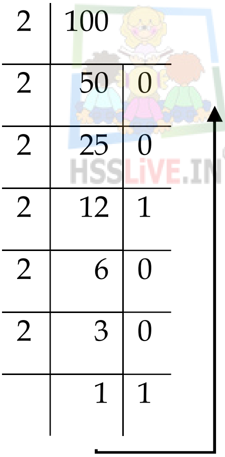
Number conversions

I. Decimal to Other Number system conversion

For converting a decimal number to other number system, divide the number continuously with the base of the number system and then take the reminders in reverse order.

1. Decimal to Binary

a. Convert 100 to binary



2	100	
2	50	0
2	25	0
2	12	1
2	6	0
2	3	0
	1	1

100 = 1100100 $_2$

2. Decimal to Octal

b. Convert 1000 to octal

8	1000	
8	125	0
8	15	5
	1	7

1000 = 1750_8

3. Decimal to Hexadecimal

c. Convert 1000 to hexadecimal

16	1000	
16	62	8
	3	14 (E)

1000 = $3E8_{16}$

II. Other Number system to Decimal conversion

To convert from other system to binary, multiply each digit by place value (power of base) and find the sum.

1. Binary to Decimal

a. Convert 111011_2 to decimal

Its positional weight is

1	1	1	0	1	1
2^5	2^4	2^3	2^2	2^1	2^0

$= 1x2^0 + 1x2^1 + 0x2^2 + 1x2^3 + 1x2^4 + 1x2^5$

$= 1x1 + 1x2 + 0 + 1x8 + 1x16 + 1x32$

$= 1 + 2 + 0 + 8 + 16 + 32$

$= 59 = 59_{10}$

2. Octal to Decimal

b. Convert 756 to Decimal

$= 6x8^0 + 5x8^1 + 7x8^2$

= 6x1 + 5x8 + 7x64

= 6 + 40 + 448

$= 494_{10}$

3. Hexadecimal to Decimal

c. Convert 9BA to Decimal

$= 10x16^0 + 11x16^1 + 9x16^2$

= 10x1 + 11x16 + 9x256

= 10 + 176 + 2304

$= 2490_{10}$

III. Octal to Binary and Binary to Octal

The following table can be used for Octal-Binary conversion

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

1. Octal to Binary

a. Convert 645_8 to Binary

Replace each digit with respective 3 digit binary code from the table

$645_8 = 110100101_2$

2. Binary to Octal

b. Convert 1100110101_2 to Octal

Make three bit groups starting from the right and replace the three bits with the corresponding octal number.

$1,100,110,101 = 1465_8$

IV. Hexadecimal to Binary and Binary to Hexadecimal

The following table can be used for Hexadecimal-Binary conversion

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

1. Hexadecimal to Binary

a. Convert $2A9F_{16}$ to Binary

$2A9F_{16} = 0010101010011111_2$

2. Binary to Hexadecimal

b. Convert 10110100110111_2 to Hexadecimal

10110100110111 = 10,1101,0011,0111
= 2 D 37
= $2D37_{16}$

V. Hexadecimal to Octal and Octal to Hexadecimal

1. Hexadecimal to Octal

First convert to Binary, and then convert to octal

a. Convert $AAE9_{16}$ to Octal

$AAE9_{16} = 1010101011101001_2$
= 1,010,101,011,101,001
= 125351_8

2. Octal to Hexadecimal

First convert to Binary, and then convert to hexadecimal

b. Convert 2345_8 to Hexadecimal

$2345_8 = 010011100101_2$
= 0100,1110,0101
= $4E5_{16}$

Number Representation

For representing positive and negative numbers in a computer, three methods are used. Those are

1. Sign and magnitude
2. 1's complement
3. 2's complement

1. Sign and magnitude

In this method, the left most bit is used for sign (+ve or -ve). If the left most bit is zero, then the number is positive. If the left most bit is one, then the number is negative.

0	+ve
1	-ve

E.g. Represent 5 and -5 using sign and magnitude method using 4 bits

5 = 0101

-5 = 1101

In this method we have two representations for zero, 0000 and 1000 (With 4 bits)

If there are n bits, then we can represent numbers from

$$\mathbf{-(2^{n-1}-1) \text{ to } +(2^{n-1}-1)}$$

ie., with 4 bits we can represent -7 to +7 $(2^{4-1}-1 = 8-1=7)$

2. 1's Complement

In this method numbers are represented as

Positive numbers - As it is (No change)

Negative numbers - Complement of the respective positive number

(Complement of 1 is 0 and complement of 0 is 1)

E.g. Represent +19 and -19 in 1's complement using 8 bits

+19 = 00010011

- 19 = 11101100

In this method also there are two representations for zero.

If there are n bits, then we can represent numbers from

$$\mathbf{-(2^{n-1}-1) \text{ to } +(2^{n-1}-1)}$$

3. 2's Complement

In 2's complement numbers are represented as

Positive numbers - As it is (No change)

Negative numbers - Adding 1 to the 1's complement of that number

E.g. Represent +19 and -19 in 1's complement using 8 bits

+19 = 00010011

- 19 = 11101101 (11101100 + 1)

In 2's complement there is only one representation for zero.

If there are n bits, then we can represent numbers from

$-(2^{n-1})$ to $+(2^{n-1} - 1)$

Comparison between different methods (Assume 8 bits are used)

	Sign and Magnitude	**1's complement**	**2's complement**
Range	-127 to +127	-127 to + 127	-128 to +127
Representation for zero	2	2	1
Total numbers	255	255	256
For representing positive number	As it is	As it is	As it is
For representing negative number	Left most bit 1 and the rest is same	Complement of +ve number	1's complement +1

Representation of Characters

Different methods are used for character representation. Those are ASCII, EBCDIC, ISCII and Unicode.

1. ASCII

ASCII stands for American Standard Code for Information Interchange. It uses 7 bits for representation. A unique number binary number is assigned to each character in this method.

E.g. ASCII code of A = 1000001

Another version of ASCII is ASCII-8 which uses 8 bits. It can represent 256 characters.

2. Unicode

It is developed by Unicode consortium. It is basically a 16 bit code. Nowadays Unicode uses more than 16 bits. It can represent all written languages in the world.

Introduction to Boolean Algebra

Boolean algebra is the branch of algebra in which the values of the variables are the truth values 'true' and 'false', usually denoted as 1 and 0, respectively. The name Boolean is given to honour British mathematician George Boole. Basic Logical operators are

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

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

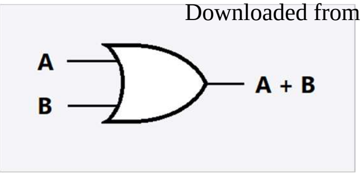
Computers perform logical operations with electronic circuits, called logic circuits. A Logic circuit is constructed with logic gates. A **logic gate** is a physical device that can perform logical operations on one or more logical inputs and produce a single output. Logic gates are implemented using diodes or transistors. Three logic gates are OR, AND and NOT.

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

OR gate

OR operator is used for logical addition. The symbol used for OR operation is +. The expression A+B is read as A OR B.

The logic gate used to implement the logical OR operation is called logical OR gate. Output of this gate will be 1 if any input is 1. The logical OR gate is depicted below.

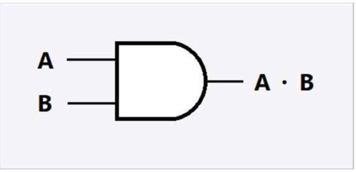


Truth table for OR operation

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

AND gate

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

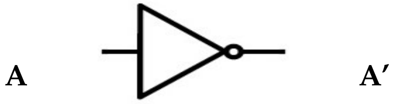


Truth table for AND operation

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

NOT gate

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



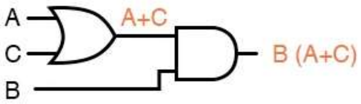
Truth table for NOT operation

A	A′
0	1
1	0

Circuit designing for Boolean expressions

Construct a logical circuit for Boolean expression B . (A + C)

Ans:



EXCERCISES

1 Score Questions

1. The base of octal number system is …….
2. How many different digits does the Hexadecimal number system contain?

3. ASCII stands for …………

4. Which number representation can represent all written languages in the world?

2 Score Questions

1. Represent -25 in 2′s complement method.

2. Explain sign and magnitude method.

3. 63_{10} = ------- $_{2.}$

4. 10110110_2 = --------$_8$

5. Explain Unicode.

6. Draw AND gate symbol.

3 Score Questions

1. Convert the decimal number 42.125 to binary.

2. 123_8 = ------$_{16}$

3. Construct a logical circuit for the Boolean expression A. (B+C)

Chapter 3

Components of the Computer System

Focus Area: Processor, Ports, Memory (RAM only with measuring units), e-Waste and disposal methods, System software (OS, Language processors compiler and interpreter), Free and open source software.

Hardware

Hardware is the tangible and visible parts of a computer.
E.g. Processor, motherboard etc.

Software

Software is a set of programs which helps to use the computer system efficiently and effectively.
E.g. Operating System, MS Office, Antivirus etc.

Humanware (Liveware)

It refers to the human who uses the computer.

Types of Hardwares.

1. Processor (CPU)

CPU controls and coordinates all operations of a computer. CPU contains millions of transistors with a silicon chip. CPU is also called as microprocessor. CPU contains registers, ALU and CU.
E.g. Intel i3, Intel i5, AMD Quad core etc.

2. Motherboard

Motherboard is a Printed Circuit Board (PCB) to which all the major components of a computer is connected.

3. Ports

Ports are used to connect external devices. Peripheral devices communicate with the motherboard through ports.

Types of Ports

i. Serial port

A serial port transmits data one bit at a time. It is very slow.

ii. Parallel port

It can transmit several bits of data simultaneously. It is faster than serial port.

iii. USB port

USB (Universal Serial Bus) is a connection that provides high speed data communication between devices. It is used for short distance communication. It is capable of supplying electric power to devices. It is used to connect devices like keyboard, mouse, printer, flash drive etc.

iv. LAN port

LAN (Local Area Network) port is used for a wired network connection. RJ45 is a standard connector for LAN connection.

v. PS/2 port

PS/2 (Personal System/2) ports are developed by IBM for connecting keyboard and mouse. This port is slow.

vi. Audio port

Audio ports are used to connect audio devices like speaker, microphone etc. Three common audio ports are line-in, line-out and mic-in.

vii. VGA Port

VGA (Video Graphics Array) port is used to connect a monitor or projector to a computer. It has 15 pins displayed in three rows.

viii. HDMI port

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

4. Memory

Memory is used to store data and instructions temporarily or permanently. Memory measuring units are given below

Binary Digit =	1 Bit
1 Nibble =	4 Bits
1 Byte =	8 Bits
1 KB (Kilo Byte)=	1024 Bytes

1MB (Mega Byte) =	1024 KB
1 GB (Giga Byte) =	1024 MB
1 TB (Terra Byte) =	1024 GB
1 PB (Peta Byte) =	1024 TB

Memory can be classified in to two, primary memory and secondary memory.

Primary Memory

Primary memory holds data and results temporarily. It is semiconductor memory. It is directly accessed by the CPU. It can send and receive high speed data. Three types of primary memory are RAM,ROM and cache.

i. RAM

RAM (Random Access Memory) is used directly by the CPU. Data can be stored and retrieved anywhere from RAM with the same speed. RAM is volatile (Contents will be lost when power is off). Capacity of RAM is 2GB, 4GB, 8GB etc.

ii. ROM

ROM(Read Only Memory) is a permanent memory. It can only be used for read operations. ROM is non-volatile. It is used to store boot-up program called BIOS. It is slower than RAM

Difference between RAM and ROM

RAM	ROM
It is faster than ROM	Slower
It stores OS, programs and data	It stores boot programs
It allows reading and writing	Allows reading only
It is volatile	Non-volatile

E-waste

E-waste refers to electronic products nearing the end of their 'useful life'. It can be discarded computers, electronics, mobile phones, television sets and refrigerators. This e-waste contains toxic materials which cause cancer, reproductive disorders and many health problems.

E-waste disposal methods

1. Reuse

It refers to second-hand use or usage after the equipment is modified. Used equipments are passed to relatives or exchanging for money or passed on to charity institutes.

2. Incineration

It is a complete combustion process. The waste is burned in specially designed incinerators at a high temperature in the range of 900 to 1000 degree Celsius.

3. Recycling

It is the process of making new products from old products. Monitors, keyboards, hard drivers, CDs, mobiles, printers, CPUs, memory chips etc. can be recycled.

4. Land filling

It is one of the most widely used methods. In this method soil is excavated and waste material is buried in it, which is covered by a thick layer of soil.

Software

Software is a set of programs that help to use computer system efficiently and effectively. The two types of software are system software and application software.

System Software

System softwares are programs designed to control the operations of a computer. It controls the operation of hardware. Also it gives services to application softwares. System softwares include Operating System and Language processors.

1. Operating System (OS)

It is the most important software in a computer. OS is an interface between user and hardware. The OS makes the system convenient to use. OS controls and co-ordinates the operations of a computer.
E.g. Windows 7, Ubuntu, DOS etc.

Functions of OS

a. Process management

Process is a program in execution. OS allocates, de-allocates and schedules processes

b. Memory management

OS keeps track of all memory location. It allocates and de-allocates memory

c. File management

OS does the file management activities such as organizing, naming, storing, retrieving, protection, recovery etc.

d. Device management

OS controls peripheral devices using device driver softwares.

2. Language processors

Language processor converts the programming language, into machine language. Three language processors are assembler, interpreter and compiler

a. Interpreter

Interpreter converts High level program to machine language line by line. If there is an error in one line, it reports the error and stops execution. E.g. The programming language BASIC is interpreted.

b. Compiler

Compiler converts high level program to machine language. It compiles the entire program and reports, the list of errors. E.g. The programming language C++ has many compilers.

Free and Open source software

Free and open source software gives the user the freedom to use, copy, change and distribute the software. Nowadays free and open source software is widely used throughout the world. Four freedoms are

Freedom 0 - Freedom to run
Freedom 1 - Freedom to change and adapt
Freedom 2 - Freedom to distribute copies
Freedom 3 - Freedom to improve and release to public

Example for free and open source software are GNU/Linux (Operating System), GIMP (GNU Image Manipulation Program), Mozilla Firefox (browser), Openoffice.org (Office suite - writer, calc, impress)

Freeware and Shareware

Difference between freeware and shareware is given below

Freeware	Shareware
Freely available	Only trial version is available for free

All the features are free	All the features are not available before purchase
Can distribute	Permission is needed to distribute

EXCERCISES

1 Score Questions

1. Set of programs which helps to use the computer efficiently is known as ….
2. ……software is the most important software in a computer.
3. The process of burning e-waste in high temperature is known as …
4. ……. Converts high level language to machine language line by line.
5. C++ is an example for …… language in a computer.

2 Score Questions

1. What is a port? Write an example for a port.
2. What is an operating system?
3. Explain different freedoms related with open source software.

3 Score Questions

1. Explain any three ports in a computer.
2. Explain different E-waste disposal methods.
3. Explain the following

   a. Interpreter b. Compiler

Chapter 4

Principles of Programming and Problem Solving

Focus Area: Phases in programming (Listing only), Debugging (Types of errors), Flowchart symbols, Development of algorithms and flowcharts to solve simple problems only (except looping).

Program

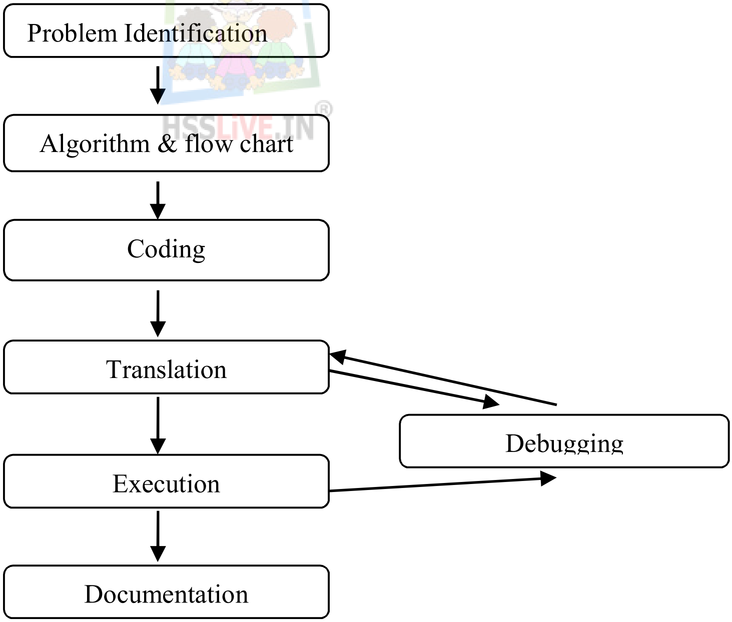
Sequence of instructions written in a computer language is known as computer program.

Approaches in problem solving

Two approaches are Top down design and Bottom up design.

Phases in Programming

The phases in programming are



1. Problem Identification

It is the first phase. This phase involves problem analysis, identification of type and quantity of data, deciding which formula to be used, what output to be obtained etc.

2. Algorithm and flowchart

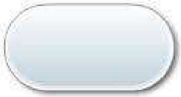
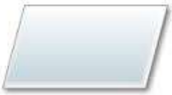
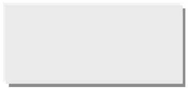
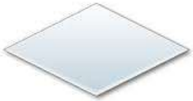
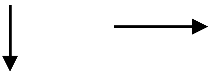
Algorithm

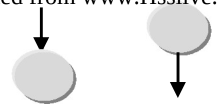
Algorithm is the step-by-step procedure to solve a problem. Arab mathematician Al-Khowarizmi is the father of the modern algorithm. An algorithm has the following characteristics

a. Input - The algorithm receives input
b. Generality - The algorithm uses variables for generality
c. Precision - The steps are precisely stated
d. Finiteness - The algorithm stops after finite steps
e. Output - The algorithm produces output

Flowchart

The pictorial representation of an algorithm is known as flowchart. A flowchart uses specific symbols and arrows for showing the sequence of operations. The symbols used are

1. Terminal (START/STOP) Symbol – Ellipse	
2. Input/Output Symbol – Parallelogram	
3. Process Symbol – Rectangle	
4. Decision Symbol – Rhombus	
5. Flow lines Symbol – Arrows	

6. Connector Symbol - Circle with a letter	

Advantages of Flowchart

a. Better communication :- program logic can be explained easily
b. Effective analysis:- Analyzing the program is easy
c. Effective synthesis:- Synthesis is easy
d. Efficient coding:- Coding is easy if a flowchart is prepared.

Limitations of Flowchart

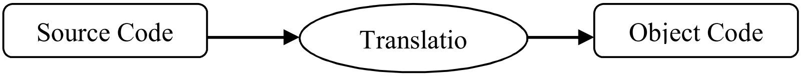
a. Time consuming
b. Changing the logic may require complete redrawing
c. No standards available

3. Coding

The process of writing the program is called as coding. Usually the programs are written in High Level Languages. The program written in HLL is known as source code.

4. Translation

Translation is the process of converting an HLL program to machine language. The compiler or interpreter is used for translation. During this step, the syntax errors of the program will be displayed. The translated code is known as object code.



5. Debugging

Debugging is the process of detecting and correcting errors (bugs). **Syntax errors** occur when the rules (syntax) of the language are not followed. E.g. Not using semicolon at the end, incorrect word, undefined term etc. Syntax errors will be detected. **Logical errors** occur when the programmer makes a logical mistake. Logical errors may not be detected by the compiler.

6. Execution

Execution is the next step after successful compilation. **Run-time errors** such as dividing by zero will be dealt in execution stage.

7. Documentation

Internal documentations are the comments in the program. **External documentations** contain instructions about the installation and usage of the program.

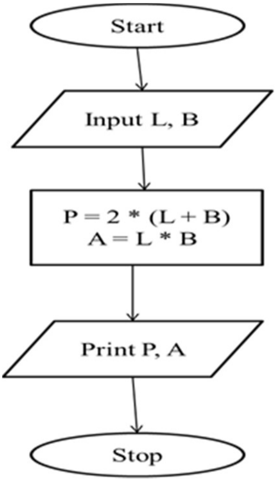
Performance evaluation of algorithms:- The performance of an algorithm is evaluated based on the time and space complexity. The algorithm which will be executed faster with minimum memory space is the best algorithm.

Questions

1. Write an algorithm and draw a flowchart to find the area and perimeter of a rectangle

Let L, B, P, A are the variables for length, breadth, perimeter and area

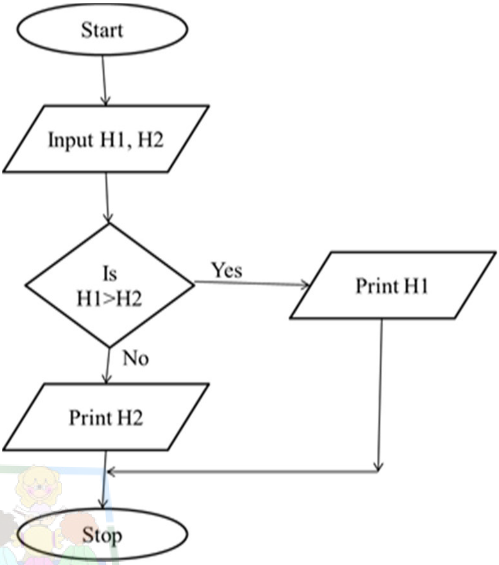
```
Step 1: Start
Step 2: Input L, B
Step 3: P = 2 * (L + B)
Step 4: A = L * B
Step 5: Print P, A
Step 6: stop
```



2. Write an algorithm and draw a flowchart to find the height of the taller one among two students

Let H1, H2 are the variables for height of the two students

```
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: Stop
```



1 Score Questions

1. Rectangle symbol in flowchart is used for ………..
2. What is a bug?
3. What is the next step after coding in a program?

2 Score Questions

1. Define program.
2. What is algorithm?
3. What is syntax error?

3 Score Questions

1. Explain phases in programming.
2. Explain three types of errors in programming.
3. Draw a flow chart to find the greatest number among two given numbers.

Chapter 5

Introduction to C++ Programming

Focus Area: Tokens and classification with examples

C++ is a powerful object oriented language. It is developed by Bjarne Stroustrup.

Character set

C++ character set includes Letters (A B C D …., a b c d …), Digits (0 1 2 …), Special characters (* # ; & { ‘ % ….), white spaces and other characters.

Tokens

Tokens are the fundamental building blocks of the program. C++ has five types of tokens. Those are keywords, identifiers, literals, punctuators and operators.

1. Keywords

Keywords are reserved words. It is also known as pre-defined words. Keywords have special meaning and it cannot be changed.
E.g. break, case, char, const, continue, do, double, else, for, goto, if, int, long, return, short, signed, sizeof, switch, unsigned, void, while etc.

2. Identifiers

Identifiers are user-defined words. It is used to name different program elements. Name of memory location, name of function, name of object etc. are identifiers. The rules for making identifiers are

a. Use letters, digits and underscore(_) only

b. The first character must be a letter or underscore

c. White spaces and special characters are not allowed

d. Keywords cannot be used

e. Identifiers are case sensitive. (Upper case letters and lower case letters are different)

E.g. for identifiers are- rank, a, sum, _num1, COUNT, product etc.

3. Literals

Literals are constants that never change their value during program run. Four types of literals are integer literals, floating point literals, character literals and string literals.

i. Integer literals

Integer Literals or integer constants are constructed only by digits without decimal point. It can have + (positive) or - (negative) symbol.
Octal numbers are represented by starting with O.
Hexadecimal numbers are represented by starting with OX.
E.g. 5000, 25, -5465, -190, O546 (Octal number), OX435 (Hexadecimal number)

ii.Floating point literals

These are numbers having fractional part. Exponential form is represented using E.
2.5 can also be written as 0.25x10^{1}. which is represented as 0.25E1
E.g. 1.25, -724.786, 45.2E6, 75.E-3, 2.25E07

iii. Character literals

A single character enclosed in single quotes is known as character literals or character constants.
E.g. ‘y’, ‘P’
Non-graphic symbols are represented by using **escape sequences (\ ).** Some non-graphic symbols represented as character constants are listed below

Character constant	Used for
\n	New line
\t	Tab (Horizontal)
\0	Null character
\a	Audible bell
\’	Single quote
\”	Double quote

iv. String literals

A sequence of one or more characters enclosed in double quotes in known as string literals.
E.g. “Hello”, “123”, “C++”

4. Punctuators

Punctuators are special symbols used in the program.

E.g. * # ( ) { } [ ] ; : ‘ “ & < >

5. Operators

Operators are used to represent some operations. The operators are applied on **operands**.

E.g. + - * / %

Symbol	Name	Symbol	Name	Symbol	Name
*	Star	#	Hash	( )	Parenthesis
{ }	Braces	[ ]	Square bracket	;	Semi colon
:	Colon	‘ ’	Single quotes	“ ”	Double quotes
< >	Angle bracket	/	Forward slash	\	Backward slash

If a + b is the operation, then a and b are operands and + is the operator.

Integrated Development Environment (IDE)

IDE provide environment for program development. Some compilers which provide IDE are Turbo C++, Geany, Borland C++ etc. C++ programs are saved with the extension **.cpp**

EXCERCISES

1 Score Questions

1. C++ is developed by ……….
2. Define tokens in C++.
3. Non graphic symbols are represented by using ……..
4. The character '\n' is used for …..
5. IDE stands for ……..

2 Score Questions

1. What is a keyword? Write an example for keyword.
2. What are the rules to be followed for making identifiers?
3. Which of the following are identifiers? Justify your answer.
   a. Num1 b. _first c. my name d. 1paper
4. What is operator? How is it related with operand?
5. What is punctuator? Write some examples.

3 Score Questions

1. Explain any three tokens used in C++.
2. Explain different types of literals used in C++.

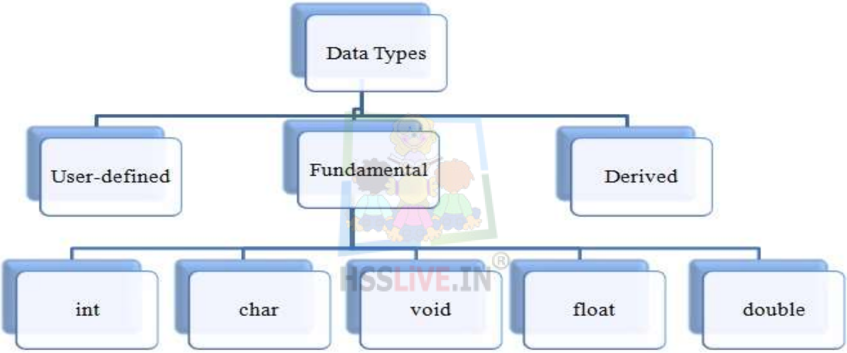
Chapter 6

Data Types and Operators

Focus Area: Fundamental data types, Variables, Operators and classifications, Type conversion, Various types of statements, Structure of C++ program.

I. Data Types

Data types are used to specify the type of the data. Data types are classified in to three, Fundamental data types, user defined data types and derived data types.



1. Fundamental data types (Built-in data types)

It is also known as pre-defined data types or built-in data types. The five fundamental data types are int, char, void, float and double.

i. int

The keyword *int* is used to represent integer numbers. Compiler allocates 4 bytes of memory for *int* data type.

ii.char

The keyword *char* is used to represent a character. The *char* data type is internally treated as integers (ASCII code of characters). Compiler allocates 1 byte of memory for char.

iii. float

The keyword *float* is used to represent floating point numbers (numbers with fractional part). Floating point numbers are stored as scientific notation. The number 1234.56 = $0.1234x10^4$ = 0.1234E4.

0.1234 is called the mantissa and 4 is the exponent. Compiler allocates 4 bytes of memory for float.

iv. double

The keyword *double* is used to represent large precision fractional numbers. Compiler allocates 8 bytes of memory for *double*.

v. void

The keyword *void* is used to represent empty data. Compiler does not allocate any bytes for void.

2. User-defined data types

It is the data type defined by users. E.g. struct, enum, class etc.

3. Derived data types

Derived data types are constructed using fundamental data types. E.g. Array, pointer, function etc.

II. Type modifiers

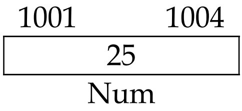
Type modifiers are used to change the size of the data type. Important type modifiers are **signed, unsigned, short** and **long**.

Since *int* has four bytes of memory, *long int* has 8 bytes of memory and *short int* has 2 bytes of memory. *double* has 8 bytes and long double has 16 bytes of memory (For gcc compiler. Memory allocation is compiler dependent).

III. Variables

Variables are the names given to memory locations. These are identifiers. Every variable has an address (L-value) in memory. The starting address is called as base address. The value stored in the variable is called as the content.

E.g. int Num = 25 ;
Here the variable is Num.
The content of the variable is 25.
The base address of the variable is 1001. .



IV. Operators

Operators are the symbols used to represent some operations.

Unary operator: - If the operator is applied on a single operand, then it is known as unary operator.
Binary operator: - If the operator is applied on two operands then it is known as binary operator.
Ternary operator: - If the operator is applied on three operands then it is known as ternary operator.
Operators can be classified in to arithmetic operators, relational operators, logical operators, input/output operators and assignment operators.

1. Arithmetic operators

The operators + (addition), - (subtraction), * (multiplication), / (division), % (modulus) are arithmetic operators.

The **modulus operator (%)** gives the remainder value of division. E.g. 10%3 = 1, 11%3=2, 12%3=0

2. Relational operators

The operators < (less than), > (greater than), <= (less than or equal to), >= (greater than or equal to), == (equal to) and != (not equal to) are relational operators. These are binary operators. The result of any relational operator will be either **True** or **False**.

3. Logical operators

The operators **&&** (logical AND), **||** (logical OR) and **!** (logical NOT) are logical operators. ! is a unary operator.

A	B	A&&B	A||B	!A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

4. Input/output operators

The **extraction** operator **>>** is the input operator. It is also known as **get from** operator

The **insertion** operator **<<** is the output operator. It is also known as **put to** operator.

5. Assignment operator

The operator = is the assignment operator. This is a binary operator. The first operand should be a variable.
E.g. a = b;   Here the value of b is stored in a.

6. Arithmetic assignment operators

Arithmetic assignment operators are used to write expressions in short form. The arithmetic assignment operators are +=, -=, *=, /= and %=. These are also known as C++ short-hands. Using short-hands makes the operations faster.

E.g. x+=10 is equivalent to x= x+10, x-=10 is equivalent to x=x-10, x*=10 is equivalent to x=x*10 etc.

7. Increment (++) and decrement (--) operators

Increment operator (++) is used to increment the integer by one.

Decrement operator (--) is used to decrement the integer by one.

8. Conditional operator

It is a ternary operator. It requires three operands. It can be used as an alternative to if-else statement. The general form is

Test expression **?** True_case code **:** False_case code **;**

E.g. if a, b and big are integers, then,

big = (a > b) ? a : b ;

Now big has the biggest value among a and b.

9. sizeof operator

sizeof is a unary operator which returns the amount of memory allocated in bytes for the operand. The syntax is

sizeof(data type) or sizeof(variable name) or sizeof(constant)

E.g. sizeof(int)

V. Type conversion

Changing the data type of one operand to another data type is known as type conversion. Two types of type conversions are implicit type conversion and explicit type conversion.

1. Implicit type conversion

Implicit type conversion is performed by the compiler. C++ converts lower type to higher type automatically. This is also known as **type promotion**.

E.g. int n = 1;
char c = 'A';
cout << n + c; //Here c is automatically converted to int for addition. Output is 66 (ASCII value of 'A' is 65)

2. Explicit type conversion

Explicit type conversion is performed explicitly by the programmer. It is also known as **type casting**.

E.g. int a=1;

float b=2.2;

cout << int (a+b); //Here a+b is explicitly converted to int. Output is 3

VI. Statements

Statements are the smallest executable unit of a programming language. Statements end with the semicolon (;). Different types of statements are

1. Declaration statements

Declaration statement is used to specify the type of the variables. The syntax for declaration statement is

data_type varibale1, variable2, variable3 ….. ;

E.g. int a,b ; // Here a and b are declared as integer type variables.

2. Assignment statements

When the assignment operator is used to assign a value to a variable, it forms an assignment statement. The syntax is

variable = constant/variable2/expression ;

E.g. a = 5; b = a, c = a + b;

3. Input statements

An input statement contains the extraction operator (>>) and the keyword **cin**. The syntax is
cin >> variable1 ;
E.g. cin >> a ;

4. Output statements

An output statement contains the insertion operator (<<) and the keyword **cout**. The syntax is

cout << variable1 ; or cout << "Any string" ;

E.g. cout << a ; or cout << "Hai" ;

VII. Cascading of I/O operators

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

VIII. Structure of a C++ program

The basic structure of a C++ program is

```
#include<iostream>
using namespace std;
int main ( )
{
    Statememts;
            :
            :
    return 0;
}
```

The first line is called pre-processor directive. *iostream* is the header file Second line is the namespace statement. *std* is the namespace identifier. Third line onwards is the main function, which is must in every program.

IX. Programs

1. To find the simple interest

```
#include<iostream>
using namespace std;
int main( )
{
        float p,n,r,si ;
        cout << “Enter the principal amount: ”;
        cin >> p ;
        cout << “Enter the number of years: ”;
        cin >> n ;
         cout << “Enter the rate of interest: ”;
        cin >> r ;
        si = p * n * r / 100 ;
        cout << “The simple interest is = ”<< si;
        return 0;
}
```

2. To find the ASCII code of a character

```
#include<iostream>
using namespace std;
int main( )
{
        char ch ;
        int asc;
        cout << "Enter the character: ";
        cin >> ch ;
        asc = ch ;
        cout << "The ASCII code is = "<< asc;
        return 0;
}
```

EXCERCISES

1 Score Questions

1. Compiler allocates ……. byte/bytes of memory for *char* data type.
2. The operator << is known as ……… operator.
3. …… is the assignment operator.
4. The keyword used for input operation is ……….
5. 15 % 4 = ……….

2 Score Questions

1. What do you mean by data types? Name different types.
2. What is a variable?
3. What is a statement?
4. What is declaration?
5. Write the three logical operators in C++.

3 Score Questions

1. Explain any three fundamental data types.
2. Explain unary, binary and ternary operators.
3. Explain different types of operators.
4. Explain different types of statements.

Chapter 7

Control statements

Focus Area: Decision making statements (if, if-else, if-else if, switch), Iteration statements (while, for, do-while) - syntax and working, (Nesting not required), Jump statements (break, continue). (No programming)

Control statements are used to control the flow of program execution. They include decision making statements and loops.

Decision making statements

Two types of decision making statements (Selection statements) are **if** and **switch**. **if** has four variants such as- if, if else, nested if and else if ladder

1. if statement

It is used to select a set of statements based on a condition. The syntax (general form) and flow chart of *if* is

Syntax	Example

if (test expression)
{
    statement block;
}
	if (score >= 18)
	    cout<< "Passed" ;

The test expression is a condition which can be relational or logical expression. The statement block is executed only if the test expression is True. If there is only a single statement in *if,* then the braces { } are not necessary.

2. if-else statement

This statement is used to include statement block for true part as well as false part. The syntax and flow chart of if else is

Syntax	Example
if (test expression)	if (score >= 18)
{	
statement block1;	cout<< "Passed";
}	
else	else
{	
statement block2;	cout<< "Failed";
}	

3. The else-if ladder

It is used for multiple branching. It is also known as else if stair case. The syntax is

Syntax	Example
if (test expression 1)	if (score >= 80)
statement block 1 ;	cout<< "A Grade" ;
else if (test expression 2)	else if (score > = 60)
statement block 2 ;	cout << "B Grade" ;
else if (test expression 3)	else if (score >= 40)
statement block 3 ;	cout << "C Grade" ;
………	else if (score >= 30)
else	cout << "D Grade" ;
statement block n ;	else
	cout << "E Grade" ;

4. Nested if

An *if* statement inside another *if* block is called nested if.

II. Switch

switch is also used for multiple branching. Its syntax is

```
switch (expression)
{
        case constant1   : statement block 1 ;
                           break;
        case constant2  : statement block 2 ;
                          break ;
                          :
                          :
        default          : statement block n ;
}
```

A program to display the day of the week, using switch statement.

```
  # include <iostream>
  using namespace std ;
  int main ( )
  {
        int day;
        cout << "Enter a number 1-7 " ;
        cin >> day;
        switch (day)
        {
              case 1 : cout << "Sunday" ;
                       break ;
              case 2 : cout << "Monday" ;
                       break ;
              case 3 : cout << "Tuesday" ;
                       break ;
              case 4 : cout << "Wednesday" ;
                       break ;
              case 5 : cout << "Thursday" ;
                       break ;
              case 6 : cout << "Friday" ;
                       break ;
              case 7 : cout << "Saturday" ;
                       break ;
              default : cout << "Invalid Choice" ;
        }
        return 0;
   }
```

The difference between switch and else-if ladder

Switch	else if ladder
Evaluates condition with equality operator only	Evaluate any relational or logical expression
Case constants must be an integer or character	Condition may include a range and float values
When no match is found, then default is executed.	When no match is found, else block is executed
break statement is required for exit from switch	Program control automatically goes out
Efficient when same variable is compared against a set of values	More flexible and versatile compared to switch.

III. The conditional operator

It is a ternary operator. It requires three operands. It can be used as an alternative to if-else statement. The general form is

Test expression **?** True_case code **:** False_case code **;**

E.g. if a, b and big are integers, then,

big = (a > b) ? a : b ;

Now big has the biggest value among a and b.

IV. Iteration statements (Loops)

Iteration (Looping) statements are used to execute a set of statements repeatedly. Four elements of a loop are

i. Initialization (Variables are initialized with first value of the loop),
ii. Test expression (It is the checking of the loop),
iii. Update statement (It modifies loop control variable) and
iv. Body of the loop (The statements to be executed repeatedly)

Three loop statements are **for, while** and **do-while**. for and while are entry controlled loops and do-while is exit controlled loop.

1. while

It is an entry controlled loop. The body of the loop will be executed only if the condition is true. The syntax is

Syntax	Example
variable initialization;	int n = 1 ;
while (testing)	while (n < = 10)
{	{
body of the loop;	cout << n << “ ”;
variable updation;	n++ ;
}	}

2. for

It is an entry controlled loop. All loop elements (initialization, test expression and update statement) are placed together in *for* statement. The syntax is

Syntax	Example
for(initialization;test;update)	for (n=1; n < =10; n++)
{	{
body of the loop ;	cout << n << “ ” ;
}	}

3. do-while

It is an <u>exit controlled</u> loop. The body of do-while will be executed at least once. Testing is done after the first execution. Its syntax is

Syntax	Example
variable initialization	int n = 1 ;
do	do
{	{
loop body;	cout << n << “ ”;
variable updation;	n++ ;
}while(testing);	} while (n < = 10);

Comparison among the loops

For	While	do-while
Entry controlled	Entry controlled	Exit controlled
Initialisation with loop definition	Initialisation before loop definition	Initialisation before loop definition
Updation with loop definition	Updation inside loop body	Updation inside loop body
No guarantee to execute at least once	No guarantee to execute at least once	Execute at least once

V. Jump statements

Jump statements are used to transfer the control from one place to another. Important jump statements are *return, goto , break* and *continue*. The function *exit()* is used to terminate a program.

break

A break statement takes the control out of the immediate enclosing loop. In a switch statement *break* causes to exit from the switch. The loop terminates its execution by break.

E.g.

```
    for (int i = 1 ; i < 10 ; i++)
    {
         if ( i = = 5 )
              break ;
         cout <<  i << “ \t ” ;
    }
```

The output of the above program would be: 1 2 3 4

continue

A continue statement is used to skip one execution of a loop. The loop is not terminated by invoking *continue*.

E.g.

```
for (int i = 1 ; i < 10 ; i++)
{
     if ( i = = 5 )
          continue ;
    cout << i << " \t " ;
 }
```

The output of the above program would be: 1 2 3 4 7 8 9

Difference between break and continue

Break	*Continue*
Used with switch or loops	Used only with loops
Takes the control outside the loop	Does not take control out of loop
Causes to terminate the loop	Skips one iteration of the loop

EXCERCISES

1 Score Questions

1. ……. is an exit controlled loop.
2. *while, do while* and …… are the three loops.
3. ……. loop will be executed at least once.

2 Score Questions

1. What are the four variants of *if* statement?
2. What is nested if?
3. Explain two jump statements.
4. Write a for loop statement to print the first 10 natural numbers.

3 Score Questions

1. What are the differences between *switch* and *else if* ladder?
2. What is a loop? What are the four elements of a loop?
3. Write differences between *for* loop *while* loop and *do while* loop.

Chapter 8

Arrays

Focus Area: Declaration, Initialisation, Accessing elements, Operations (listing only with concept), Traversal operation with simple program.

Array is a fixed number of elements of the same type stored sequentially in memory with a single name. An integer array holds integers only, a character array holds characters only and so on. Each element can be accessed through **index** number or **subscript**.

Syntax for array declaration

Data-type arrayName[size] ;

E.g. int a[5] ; (This declares an array which can hold 5 integers)

Each elements in an array is indexed as follows

a[0]	a [1]	a [2]	a [3]	a [4]

Elements can be stored in to this array as

a [0] = 6 ; a [1] = 10; etc.

The memory space allotted for an array can be calculated as

Total-Bytes = sizeof(Data-type) x size of array.

In the above example, Total bytes = sizeof(int) x 5 = 4x5 = 20 Bytes.

(int needs 4 bytes in GCC compiler)

Initialization (giving values) of an array cab be done as

int a[5] = { 9, 6, 10, 20,8 } ;

or int a[ ] = { 9, 6, 10, 20,8 } ;

Traversal: - Accessing each elements of an array at least once is known as traversal of the array.

The program below finds the sum of the elements of an array

```
#include<iostream>
using namespace std;
int main()
{
```

```
            int a[5], i, sum=0;
            cout << ”Enter the elements of the array”;
            for(i=0; i<5; i++)
                cin >> a[i];
            for(i=0; i<5; i++)
                sum = sum+a[i];
            cout << ”Sum of the elements of the array is”<<sum;
            return 0;
}
```

Write a C++ program to create an array of N numbers and find the average of the elements.

```
#include<iostream>
using namespace std;
int main()
{
        int a[100], N, avg, i, sum = 0;
        cout << “Enter number of elements”;
        cin >> N ;
        cout<< “Enter the elements of the Array”;
        for(i=0 ; i < N ; i++)
            cin>> a[i];
        for(i = 0 ; i < N ; i++ )
            sum=sum+a[i] ;
        avg = sum / N ;
        cout << “The average of the elements = ” << avg ;
        return 0 ;
}
```

EXCERCISES

1. Define an array
2. What is traversal in an aray?
3. Write the syntax for array declaration.
4. Write an array declaration to store 5 marks of a student.

Chapter 9

String Handling and I/O Functions

Focus Area: Array declaration for string and initialisation, Input/Output operations, Use of get(), getline(), put(), write() functions. (No programming)

String Handling using arrays

String can be stored using character arrays.
E.g. char str[10]= "Sachin";

A **null character '\0'** is automatically inserted at the end of a string as a delimiter character. The null character helps to identify the end of a string. The memory allocation is as follows

S	a	c	h	I	N	\0			
0	1	2	3	4	5	6	7	8	9

Only one word can be read if **cin** is used for reading a string.

E.g.

```
    char str[10];
    cin >> str;
```

If the input given is "Sachin Tendulkar", then only Sachin will be saved on the array 'str'. Stream functions can be used to read and write strings.

Stream Functions

Stream functions are more robust C++ functions for character and string operations. The header file to include is **iostream**. These functions allow a stream of bytes to flow between memory and objects (like keyboard). The object cin and '.' (dot operator) need to be used for input operations. The object cout and '.' need to be used for output operations. Important stream functions are get(), getline(), put() and write().

a. get()

This can accept a character or a string through keyboard. For accepting string array needs to be given as argument.

E.g. 1.

char ch;
cin.get (ch); // This accepts a character to the variable ch

E.g. 2. char str[20];
cin.get(str,10) // This accepts a string of maximum 10 characters

b. getline()

This accepts a string through keyboard. The delimiter (end character) will be Enter key. It can have one, two or three arguments. Third argument is to set any other delimiter character.

E.g. char str [20] ;
cin.getline (str, 10) ; // This accepts a string of maximum 10 characters
cin.getline (str, 10, 'z') ; // This accepts a string of max 10 characters or up to the letter 'z'.

c. put()

This is used to display a character to the monitor. The object cout is used.

E.g. char ch = 'K'
cout.put (ch); // Displays 'K' to the monitor

d. write()

This is used to display a string to the monitor. It can give one or two arguments. First argument is a character array and the second argument can specify how many characters to be displayed.

E.g. char str[50] = "This is to check the stream function" ;
cout.write (str, 16) ; //This will display "This is to check"

EXCERCISES

1. A …… character is inserted automatically at the end of a string.
2. Write the use of get() function in C++
3. Explain any two stream functions in C++.

Chapter 10

Functions

Focus Area: Modular programming and merits, Predefined functions (string, mathematical, character), User-defined functions (Syntax, Concept of arguments and return value). (No programming)

Modular programming

The process of subdividing a program in to separate sub programs or modules (functions) is known as modularization. Modular programming aims to achieve modularization

Advantages of modular programming

1. **Reduces program size**

Repetition of same expression can be avoided by using modules.

E.g. To find $\sqrt{x} + \sqrt{y}$, The module written for finding square root of x can be reused for y.

2. **Reduces chances of error**

Errors can be easily identified as they are localized to a function

3. **Reduces programming complexity**

Code for each module is short, simple and easy to understand.

4. **Improves Reusability**

A function once written can be reused later, which saves time and energy.

E.g. calling and using of predefined functions.

Disadvantages

1. **Breaking of the problem is difficult**

Dividing a larger program in to sub-modules is sometimes difficult.

2. **Dependency should be avoided**

When dividing in to different modules, each module should be independent of others. Otherwise it would be difficult to make error-free codes.

Functions

A function is a group of statements that together perform a task. All C++ programs have at least one function, which is **main( )** function. The syntax of a function is

```
return_type function_name( parameter list )
{
                body of the function
}
```

return_type can be any valid data type, such as int, char, float, void and double.
function_name is any name (identifier) given by the user
Parameter/Argument is variables preceded by data_types – (parameters are optional)
body of the function contains statements to perform different tasks.

E.g.

```
void show( )
{
     cout << "Hello, I am inside the function";
}
```

Function header

The head part of a function is called function header, which contains

- return_type function_name(parameter list)

E.g. int sum(int a, int b)

Function prototype

Function prototype is a declaration of a function that specifies function name, return type and number and type of parameters. Function prototype is necessary if the function is defined after main( ).

Syntax of a function prototype,

return_type function name (data types of parameters) ;

E.g. int sum (int , int) ;

Arguments (Parameters)

The variables in the parenthesis of a function header are called as arguments.

E.g. int sum(int a , int b) - Here a & b are parameters or arguments.

Function definition and Function call

Invoking a function is known as function call. Defining a function is known as function definition. Function definition includes function header and function body.

Formal and actual arguments

The variables used in the function definition are known as formal arguments. Variables or constants used in the function call are known as actual arguments.

E.g.

```
#include <iostream>
int sum(int, int ); //          Function prototype
int main( )
{
                              Actual arguments ( x and y)
     int x=5, y=10, z;

     z = sum(x,y);  //          Function call  [ sum(x,y) ]
     cout << “The sum of a and b is : ”<< c ;
     return 0;
}
                                 Formal arguments ( a and b)
int sum(int a, int b)

{
      return a+b ;                  Function definition
}
```

Types of functions	
Pre-defined functions	**User defined functions**

[These are built-in functions, which is available on C++ library. For using this we need to include the corresponding header file]
E.g. getchar( )
	These are defined by users.
	E.g. int square (int a)
	{
	return a*a ;
	}

I. Pre-defined Functions

There are so many pre-defined functions. Some of them are

1. String functions
2. Mathematical functions
3. Character functions

1. String functions

There are many different string manipulation functions under the header file **cstring**. Those are strlen(), strcpy(), strcat(), strcmp(), strcmpi().

a. strlen()

This is used to find the length of a string. It returns an integer value. Syntax is

int strlen (string) ;

E.g. char str [ ] = "Hello dear"

int p = **strlen**(str) ; // Now p has the value 10.

b. strcpy()

This is used to copy one string in to another. The second argument copies to the first argument. Syntax is strcpy(string1, string2) ; // string2 copies to string1

E.g. char s1[20], s2[20] = "Hello" ;

strcpy(s1 , s2**)** ;

cout << s1 ; // Displays "Hello"

c. strcat()

This is used to append one string in to another. The second string is appended to the first string. Syntax is strcat(string1, string2) ;

E.g. char s1[20] = "Hello " , s2[20] = "Dear" ;

strcat(s1,s2**)** ;

cout << s1 ; // Displays "Hello Dear"

d. strcmp()

This is used to compare two strings. For comparison alphabetical order is considered. Returns 0,+ve value or -ve value. Syntax is

strcmp (string1, string2)

It returns 0 – if both strings are same.

It returns +ve – if string1 is alphabetically higher

It returns -ve – if string1 is alphabetically lower

E.g. char s1[ ] = " apple" , s2[ ] = "banana" ;

int p = **strcmp(**s1,s2**)** ; // Now p has +ve value as s1 is alphabetically higher.

e. strcmpi()

This is also used to compare two strings. This function will ignore cases. Both uppercase and lowercase letters will be considered as same for comparison. This returns 0,+ve value or –ve value.

E.g. char s1[ ] = " apple" , s2[ ] = "APPLE" ;

```
int p = strcmpi(s1,s2) ; // Now p has the value 0.
```

2. Mathematical functions

Functions are available for mathematical operations. The header file to include is **cmath**. Some common mathematical functions are abs(), sqrt() and pow().

a. abs()

This is used to find the absolute value of an integer. It takes an integer as argument and return the absolute value. Syntax is

```
int abs(int) ;
```

E.g.

```
int n = -5 ;
cout<< abs(n) ; // Displays 5
```

b. sqrt()

This is used to find the square root of a number. The argument can be int, float or double. The function return the square root. Syntax is

```
double sqrt( double ) ;
```

E.g.

```
int n = 100 ;
float r = sqrt ( n ) ;
cout<< r ; // Displays 10
```

c. pow()

This is used to find the power of a number. This function takes two arguments and returns a double . The syntax is

```
double  pow( double, double) ;
```

E.g.

```
int x = 2  , y = 5 , z ;
z = pow (x, y) ;
cout << z ; // Displays 32 ( That is 2^5 )
```

3. Character functions

These functions are used to perform various operations on characters. The header file to include is **cctype**.

a. isupper( )

This is used to check whether a character is upper case or not. It returns 1 if true, and return 0 if false. Syntax is

```
int isupper (char c) ;
```

E.g. int n = isupper ('A'); // Now n has the value 1.

b. islower( )

This is used to check whether a character is lower case or not. It returns 1 if true, and return 0 if false. Syntax is

```
int islower (char c) ;
```

E.g. int n = islower ('A'); // Now n has the value 0.

c. isalpha( )

This is used to check whether a character is alphabet or not. It returns 1 if true, and return 0 if false. Syntax is

```
int isalpha (char c) ;
```

E.g. int n = isalpha ('A'); // Now n has the value 1.

d. isdigit( )

This is used to check whether a character is digit or not. It returns 1 if true, and return 0 if false. Syntax is

```
int isdigit (char c) ;
```

E.g. int n = isdigit ('5'); // Now n has the value 1.

e. isalnum( )

This is used to check whether a character is alphanumeric or not. It returns 1 if true, and return 0 if false. Syntax is, int isalnum(char c) ;

E.g. int n = isalnum (‘*’); // Now n has the value 0.

f. toupper( )

This function is used to convert a character to its upper case. It returns the upper case letter. The syntax is,

char toupper(char c)

E.g. char c = toupper (‘a’) ; // Now c has the value ‘A’

g. tolower( )

This function is used to convert a character to its lower case. It returns the lower case letter. The syntax is,

char tolower(char c)

E.g. char c = tolower (‘a’) ; // Now c has the value ‘A’

User defined functions

These functions are defined by users. No need to include any header files as the function is defined in the program itself.

E.g.

```
#include<iostream>
 int area( int a, int b)
{
        int c = a*b;
        return c;
}
int main( )
{
         int l, b , x;
         cout << “Enter the length and breadth of a rectangle: ” ;
         cin >> l >> b ;
         x = area ( l , b );
         cout << “The area of the rectangle is: ”<< x ;
         return 0 ;
```

}

In the above program the function area ( ) is a user defined function.

EXCERCISES

1 Score Questions

1. Define a function
2. Write any two mathematical functions
3. The header file for character functions is …………..
4. What will be the output of the statement, cout << strlen("I am happy"); ?

2 Score Questions

1. What are the advantages of modular programming?
2. What is function prototype?

3 Write differences between formal and actual arguments.

4 What is function definition and function call?

5 What are the differences between strcmp( ) and strcmpi( )?

3 Marks Questions

4. List out different pre-defined functions
5. Explain different mathematical functions
6. What are different character functions?
7. Identify the built-in functions required for the following cases
   a. To check whether a given character is digit or not
   b. To find the square root of 81
   c. To combine strings "Sch" and "ool" to make it as "School"
   d. To convert the letter 'K' to k.

Chapter 11

Computer Networks

Focus Area: Advantages of network, Key terms (Bandwidth, noise, node), Communication devices (switch, router, gateway, bridge, modem), Network topologies, Identification of computers over network (MAC, IP)

Computer Network is a group of computers and other devices connected together through a communication medium. In a computer network computers can communicate, and share data and resources.

Advantages of network

1. Resource sharing

Resources can be shared using network. Resources may be hardware (hard disk, printer, scanner etc.) or software (application software, anti-virus etc.).

2. Price-performance ratio

Cost of software will be reduced by sharing software without affecting performance.

3. Communication

Communication is easy and faster using networks. E-mail, chat, video conference etc. are examples.

4. Reliability

Information can be backed up in multiple computers using network. It improves reliability.

5. Scalability

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

Some network terms

1. Bandwidth: - It means the amount of data that can be sent over a specific connection in a given amount of time. It is measured in bits per second (bps).

2. Noise: - Unwanted electrical or electromagnetic energy that lowers the quality of data signals is called noise.

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

Data communication devices

1. **Network Interface Card (NIC)**

It is the hardware interface between a computer and a network. It can be a separate circuit board or integrated with the motherboard. NIC can transfer data at a speed of 1 Gbps.

2. **Hub**

It is a small, simple, passive and inexpensive device used to connect computers of the same network. Hub transmits data packets to all other computers connected to it (Broadcasting). So it increases the network traffic and reduces effective bandwidth.

3. **Switch**

Switch is an intelligent device which transmits the data only to the intended node. Switch performs better than a hub since it generates less traffic.

4. **Repeater**

A repeater is used to regenerate incoming electrical, wireless or optical signals through a communication medium. It strengthens the incoming signal and retransmit them to the destination.

5. **Bridge**

Bridge is used to connect different segments of a network. A network can be split into different segments and can be interconnected with a bridge. This reduces network traffic.

6. **Router**

Router is an intelligent device used to interconnect two networks of the same type using the same protocol. It can find the best path for data packets to travel and reduces network traffic.

7. **Gateway** Gateway is used to interconnect two different networks having different protocols. Its operations are similar to a router. It can find the best path for packets reach to the destination.

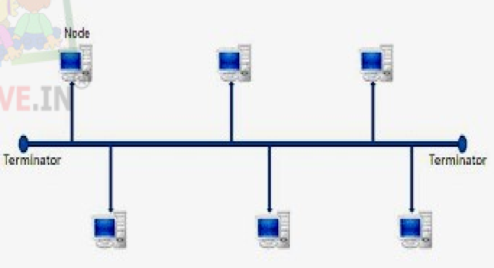
Network topologies

The way in which the nodes are connected to form a network is called a topology. Four types of topologies are

1. Bus

In bus topology, all the nodes are connected to a main cable called bus. A small device called terminator is attached at each end of the bus. Characteristics of a bus are

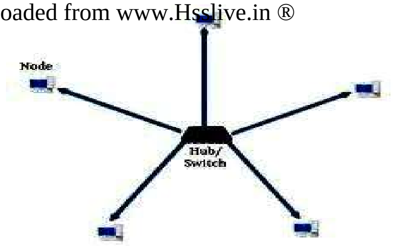
- Easy to install
- Requires less cable length
- Inexpensive
- Failure of node does not affect the network
- Failure of bus leads to network break down
- Only one node can transmit data at a time



2. Star

In star topology each node is directly connected to a hub or switch. The message is passing to the hub and it broadcast the message. Its characteristics are

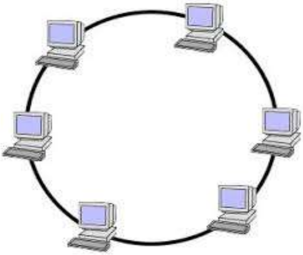
- More efficient compared to bus
- Easy to install
- Requires more cable length
- Failure of node does not affect the network
- Failure of hub leads to network break down
- Easy to expand



3. Ring

In ring topology all nodes are connected to a cable ring. Data travels only in one direction. Its characteristics are

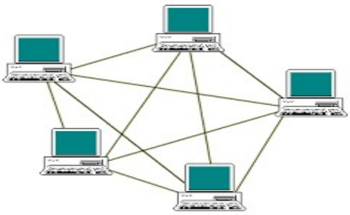
- Requires less cable length
- Inexpensive
- Failure of a node leads to network break down
- Addition of nodes is difficult
- No signal amplification is required



4. Mesh

In mesh topology every node is connected to all other nodes. So there are multiple path between nodes. Its characteristics are

- Most reliable network topology
- Network will not fail, when one path fails
- Requires more cable length



- Expensive and difficult to manage

Identification of computers over a network

MAC address and IP address are used to identify the nodes in a network

1. MAC address

MAC (Media Access Control) address is a unique 12 digit hexadecimal address assigned to each NIC. MAC addresses are usually written in the format MM : MM : MM : SS : SS : SS . The first half contains the ID number of the manufacturer and the second half represents the serial number of the NIC.

2. IP address

IP address is a unique 4 part numeric address assigned to each node on a network for identification. IP addresses are represented as dotted decimal number. E.g. 192.168.1.1 (Each number can be between 0 and 255). The two versions for IP addresses are IPv4 and IPv6. IPv4 uses 32 bit and IPv6 uses 128 bit.

EXCERCISES

1 Score Questions

1. ……. is used to interconnect two different networks having different protocols.
2. HTTP stands for ……….

2 Score Questions

1. What are the advantages of network?
2. What is the difference between hub and switch?
3. What is IP address?

3 Score Questions

1. Explain different network topologies.
2. Explain the following
   a. Repeater b. Bridge c. Router

Chapter 12

Internet and Mobile Computing

Focus Area: Services on Internet (Working procedure is not required), Cyber security (Computer virus, Trojan horse, hacking, phishing).

Internet

Internet is a global system of interconnected networks providing millions of computers worldwide to connect and exchange information. ARPANET is the first WAN (Wide Area Network). Vinton Gray Cerf is considered as the father of Internet. Tim Berners-Lee is the inventor of world wide web.

Services on Internet

1. World Wide Web (WWW)

World Wide Web is a system of interlinked hypertext documents which can be accessed through Internet using a URL.

Browser: - Browser is a software used to access the web pages in the World Wide Web. It is capable of displaying text, images, links, videos, sounds, scripts etc. Examples for browsers are Google Chrome, Internet Explorer, Mozilla Firefox, Opera, Safari etc.

Browsing: - Browsing is the process of accessing web pages.

2. Search engines

Search engine is a software system that search documents on the World Wide Web for specific keywords and returns a list of matching documents. Search engine uses web crawlers or spiders or robots to search the web. E.g. Google, Bing, Yahoo etc.

3. E-mail

E-mail or Electronic mail is a system of sending messages electronically from one computer to another via a network. An e-mail address consists of two parts separated by @ symbol. The first part is the username and the second part is the domain name of e-mail server. E.g. scertkerala@gmail.com

a. Sections of an e-mail

An e-mail has the following sections

To - Addresses of the recipients are specified here

Cc - Carbon copy is to provide address of the secondary recipients

Bcc - Blind carbon copy is to provide addresses of third recipients and the primary and secondary recipients cannot see the bcc addresses

Subject - It is to provide a meaningful subject/heading

Content - Messages are typed here

b. Advantages of using e-mail

Speed: - E-mail is delivered instantly to any location across the globe. E-mail can be sent to multiple users

Easy to use: - Organizing, saving and processing e-mails are very easy

Attaching is possible: - Pictures, files, documents etc. can be attached with e-mail

Environment friendly: - E-mails do not use paper and it save trees

Cost-effective: - Sending e-mail is inexpensive

Anywhere anytime availability: - E-mails are available from anywhere at any time.

4. Social media

a. Classification of social media

1. Internet forums: - It is online discussion websites where people can engage in conversations and find solutions in the form of messages. Each discussion on a topic is called a thread. E.g. Ubuntu forum

2. Social blogs: - A blog is a discussion or informational website consisting of posts in a topic and owned by an individual or a group of individuals. Websites like blogger.com offers blogging facility

3. Micro blogs: - It allows users to exchange short sentences, images or video links. It can be used for expressing personal opinion. E.g. Twitter.com

4. Wikis: - It allows people to add or edit content about a topic. It is a content management system. Editing done by users are monitored by other editors, so that errors can be removed immediately. E.g. Wikipedia.org

5. Social networks: - This allows people to connect with other people. It allows posting and sharing or texts, images, videos etc. It influences public opinion. E.g. facebook.com, linkedin.com

6. Content communities: - These are websites that organize and share contents like photos and videos. E.g. Youtube.com is a popular video sharing website.

b. Advantages of social media

- **Bring people together:** - Social media allows people to find lost childhood friends and also to make new friends
- **Plan and organize events:** - It help users to organize and participate in events
- **Business promotion:** - It allows businesses to connect with customers, make marketing campaigns, advertise etc.
- **Social skills:** - It allows people to express their views over a particular issue and become an agent for social change

c. Limitations of social media

- **Intrusion to privacy:** - The personal information like name, location, e-mail address and age of users can be used for illegal activities.
- **Addiction:** - Addiction to this sites wastes valuable time. It will negatively affect our mental states which may lead to depression. Students may lose concentration in studies.

- **Spread rumors:** - Social media spread the news quickly. A wrong information may spread quickly in this sites

d. Best practices for social media interaction

- Avoid unnecessary uploading of personal data like e-mail address, phone number, address, pictures etc.
- Set time schedule for using this websites
- Avoid posting wrong or misleading content on websites
- Set privacy levels (private, friends, public) such that you know exactly who can see your posts.

Cyber security

Security to the network is important because data can be lost, privacy can be violated and usual work can be interrupted by threats. Some common threats of computer network is discussed below.

1. Virus

A virus is a program that attaches itself to another executable file. It spread from one computer to another without users' knowledge. A virus may corrupt or delete data on the affected computer. Viruses spread through USB drives, file sharing, e-mail etc.

2. Worm

It is a standalone malware program that replicates itself. It spread from one computer to another on its own. A worm does not need another program to propagate.

3. Trojan horse

It will appear to be useful software, but will damage the computer by creating a backdoor on the computer. Trojans do not replicate.

4. Spams

These are junk mails indiscriminately sent to promote a product or service. Most email service providers provide email filters which separates spams from genuine mails.

5. Hacking

Hacking is a technical effort to compromise the security of a network. It is performed with good intention (white hats - ethical hacking) or bad intention (black hats)

6. Phishing

It is the process of stealing information (spoofing) by creating duplicate websites which looks almost exactly as the original web site.

EXCERCISES

1 Score Questions

1. Who is the father of internet?
2. Write the format of an email address.

2 Score Questions

1. What is internet?
2. Write advantages and disadvantages of social media.
3. Write best practices for social media interaction.

3 Score Questions

1. What is email? What are the advantages of email?
2. Explain different classification of social media.
3. Explain the following

   a. Virus b. Trojan horse c. Hacking