

CHAPTER - 1 INTRODUCTION TO COMPUTERS

Section - B : Very Short Answers

1. What is a computer?

- Computer is an Electronic Machine, capable of performing basic operations like addition, multiplication, division etc.
- Computer accepts data as input, process it, produce output and stores it for future.

2. Distinguish between data and information.

Data	Information
Data is a collection of facts from which the information may be derived	Information is a collection of facts from the conclusions can be drawn
Example: 134, 16, Kavitha	Example: Kavitha is 16 years old

3. What are the components of a CPU?

The Components of CPU are:

- Control Unit
- Arithmetic and Logic Unit (ALU)
- Memory Unit.

4. What is the function of an ALU?

- The ALU is a part of CPU where various computing functions are performed on data. ALU performs arithmetic operations such as addition, subtraction, multiplication, division and logical operations.

5. Write the functions of control unit.

- The control unit controls the flow of data between the CPU, memory I/O devices. It also controls the entire operations of a computer.

6. What is the function of memory?

- Memory enables the computer to store the program. The memory unit is of two types:
 1. Primary memory
 2. Secondary memory
- The primary memory is used to store data temporarily and secondary memory stores the data permanently.

7. Differentiate Input and output unit.

Input unit	Output unit
An input device feeds information to a computer system for processing	An output unit is any hardware component that conveys information to users understandable format.
Example: keyboard, mouse	Example: printer, monitor

8. Distinguish Primary and Secondary memory.

Primary Memory	Secondary Memory
The primary memory is used to temporarily store the programs and data when the instructions are ready to execute	The secondary memory is used to store the data permanently.
The primary memory is volatile, that is, the content is lost when the power supply is switched off.	The secondary memory is non-volatile, that is, the content is available even after the power supply is switched off.
Example: RAM	Example: Hard disk, CD-ROM and DVD-ROM

Section - C : Short Answers

1. What are the characteristics of a computer?

The Basic characteristics of the computers are:

- High speed
- Stores Huge Data
- They do millions of different tasks
- Very versatile
- Accurate
- Diligence

2. Write the applications of computer.

- Banking
- Insurance
- Education
- Marketing
- Health care
- Engineering design
- Communication
- Government
- Weather forecasting

3. What is an input device? Give two examples.

- Input Device is a peripheral device used to provide data and control signals to an information processing system.
- Example: Keyboard, Mouse, Scanners, Digital cameras, Joysticks.

4. Name any three output devices.

- The three output devices are
 - (a) Monitor
 - (b) Printers
 - (c) Multimedia Projector.

5. Differentiate optical and Laser mouse

Optical Mouse:

- It is less sensitive towards the surface
- Tracking power is less
- Old technology
- This uses the light source instead of ball to judge the motion of the pointer
- Optical mouse has 3 buttons

Laser Mouse:

- Highly sensitive
- Tracking power is more
- Latest technology
- This uses Laser light
- No. of buttons will vary from 3 to many

6. Write short note on impact printer.

- Impact printer refers to the class of printers that work by banging a head or needle against an ink ribbon to make a mark on the paper.
- Impact printers are very noisy.
- These printers can print on multi-part.
- Example: Dot Matrix, Line printers.

7. Write the characteristics of sixth generation.

- Natural Language Processing
- Development of Robotics
- Parallel and Distributed computing
- Artificial Intelligence
- Development of Robotics

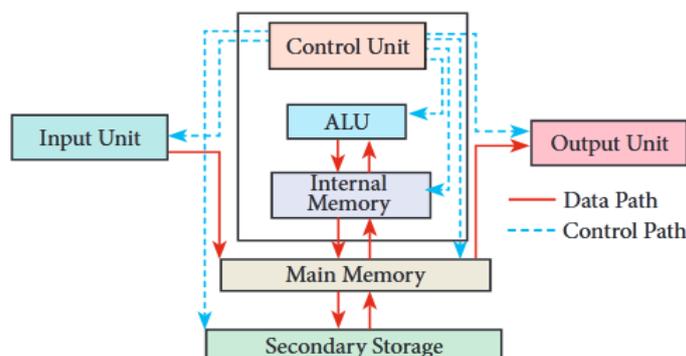
8. Write the significant features of monitor.

- A computer monitor is an output device which displays information in pictorial form.
- Monitors can be of monochrome which displays text or images in Black and white (or) in color.
- Different types of monitors are CRT, LCD and LED.

Section - D : Explain in Detail

1. Explain the basic components of a computer with a neat diagram.

The Basic components are I/O Unit, CPU, ALU, Control Unit, Storage Unit.



Input unit:

- Input unit is used to feed any form of data to the computer. It can be stored in the memory unit for further processing.
- Ex. Keyboard, Mouse, Scanner, Retinal scanner, Track ball.

Central processing unit:

- CPU is the major component and it controls the operation of all other components such as memory, input and output units.
- The CPU has three components namely control unit, Arithmetic and logic unit (ALU) and memory unit.

Arithmetic and logic unit:

- The ALU is a part of the CPU where arithmetic operations like addition, subtraction, multiplication, divisions and logical operations will takes place.

Control unit:

- The control unit controls the flow of data between the CPU, memory and I/O devices. It also controls the entire operation of a computer.

Output unit:

- An output device is a hardware component that conveys information to the user in an understandable form.
- Example: Monitor, printer, plotter

Memory unit:

- The memory stores everything that computer works with.
- The memory unit is of two types namely primary memory and secondary memory.
- The primary memory stores the data and instructions temporarily. Whereas the secondary memory stores the data permanently.
- The primary memory is volatile and secondary memory is non – volatile.

2. Discuss the various generations of computers.**First Generation (1940 – 1956) Vacuum tubes**

- Big in size
- Consumed more power
- Malfunction due to overheat
- Machine Language was used

Second Generation (1956 – 1964) Transistors

- Smaller compared to First Generation
- Generated Less Heat
- Consumed less power compared to first generation
- Punched cards were used
- Machine language as well as Assembly language was used.

Third Generation (1964 – 1971) Integrated Circuits (IC)

- Computers were smaller, faster and more reliable
- Consumed less power
- High Level Languages were used

Fourth Generation (1971 – 1980) Microprocessor Very Large Scale Integrated Circuits(VLSI)

- Smaller and Faster
- Microcomputer series such as IBM and APPLE were developed
- Portable Computers were introduced.

Fifth Generation (1980 – till date) Ultra Large Scale Integration (ULSI)

- Parallel Processing
- Super conductors
- Computers size was drastically reduced.
- Can recognise Images and Graphics
- Introduction of Artificial Intelligence and Expert Systems

Sixth Generation In future

- Parallel and Distributed computing
- Computers have become smarter, faster and smaller
- Development of robotics
- Natural Language Processing
- Development of Voice Recognition Software

3. Explain the following

- (a) Inkjet Printer (b) Multimedia projector (c) Bar code / QR code Reader

(a) Inkjet Printer:

- Inkjet printer use colour cartridges which combined magenta, yellow and cyan inks to create colour tones.
- A black cartridge is used for monochrome output. Inkjet printer works by spraying ionised ink at a sheet of paper.
- The speed of inkjet printers generally range from 1 – 20 ppm (page per minute).

(b) Multimedia Projector:

- Multimedia projectors are used to produce computer output on a big screen.
- These are used to display presentations in meeting halls or in class rooms.

(c) Barcode/ QR code Reader:

- A Bar code is a pattern printed in lines of different thickness.
- The Bar code reader scans the information on the bar codes transmits to the computer for further processing.
- The system gives fast and error free entry of information into the computer.

Quick response (QR) code:

- The QR code is the two dimension bar code which can be read by a camera and processed to interpret the image.

CHAPTER - 2 NUMBER SYSTEM

Section-B : Very Short Answers

1. What is data?

- The term data comes from the word datum which means a raw fact. The data is a fact about people, places or some objects.
Example: Rajesh, 16, XI

2. Write the 1's complement procedure.

- Step 1 : convert given decimal number into Binary
- Step 2 : if the binary bit contains 8 bits if less add 0 at the left most bit, to make it as 8 bits.
- Step 3 : Invert all the bits, (i.e.) change 1 as 0 and 0 as 1.

3. Convert $(46)_{10}$ into Binary number

2	46	
2	23 - 0	↑
2	11 - 1	
2	5 - 1	
2	2 - 1	
1	- 0	

$(46)_{10} = (101110)_2$

4. We cannot find 1's complement for $(28)_{10}$. State reason.

- Since it is a positive number. 1 's complement will come only for negative number.

5. List the encoding systems for characters in memory.

- BCD – Binary Coded Decimal
- EBCDIC – Extended Binary Coded Decimal Interchange Code
- ASCII – American Standard Code for Information Interchange
- Unicode
- ISCII – Indian standard code for Information interchange

Section - C : Short Answers

1. What is radix of a number system? Give example.

- The radix refers to the base of a number system: the total number of possible digits.
- Example : Binary (Base 2) , Decimal (Base 10) , Octal (Base 8) , Hexadecimal (Base 16)

2. Write note on binary number system.

- In Binary Number System, there are only two digits namely 0 and 1. The numbers in the binary system are represented to the base 2.
- In the Binary Number, the left most bit is the Most Significant Bit (MSB) and the right most bit is the Least Significant Bit (LSB).

- MSB has largest positional weight and LSB has smallest positional weight.

Example : 110010_2

3. Convert $(150)_{10}$ into Binary, then convert that Binary number to Octal.

Step 1: Change it to Binary.

2	150	
2	75 - 0	↑
2	37 - 1	
2	18 - 1	
2	9 - 0	
2	4 - 1	
2	2 - 0	
1	1 - 0	

$$(150)_{10} = 10010110$$

Step - 2: Change it to Octal.

$$\begin{array}{ccc} 010 & 010 & 110 \\ \hline 2 & 2 & 6 \end{array}$$

$$10010110_2 = 226_8$$

4. Write short note on ISCII.

- ISCII is the system of handling the character of Indian local languages.
- This as a 8-bit coding system.
- Therefore it can handle 256 (2^8) characters.
- This system is formulated by the department of Electronics in India in the year 1986- 88 and recognized by Bureau of Indian Standards (BIS).
- Now this coding system is integrated with Unicode.

5. Add (a) $-22_{10} + 15_{10}$ (b) $20_{10} + 25_{10}$

(a) $-22_{10} + 15_{10}$

2	22	
2	11 - 0	↑
2	5 - 1	
2	2 - 1	
1	1 - 0	

$$22_{10} = 10110_2$$

Finding the binary value of -22_{10}

8-bit for of 22_{10}	=	00010110
1's complement	=	11101001
Add 1	=	1
2's complement	=	11101010

2	15	
2	7 - 1	↑
2	3 - 1	
1	1 - 0	

$$15_{10} = 1111_2$$

$$\begin{array}{r} -22_{10} + 15_{10} = \quad 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ + \quad 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1 \end{array}$$

(b) $20_{10} + 25_{10}$

2	20	
2	10 - 0	↑
2	5 - 0	
2	2 - 1	
1	1 - 0	

$$20_{10} = 10100_2$$

2	25	
2	12 - 1	↑
2	6 - 0	
2	3 - 0	
1	1 - 1	

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

$$\begin{array}{r} 20_{10} + 25_{10} = \quad 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0 \\ + \quad 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1 \\ \hline 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1 \end{array}$$

Section - D : Explain in Detail

1. (a) Write the procedure to convert fractional Decimal to Binary. (b) Convert $(98.46)_{10}$ to Binary

(a) Procedure to convert fractional Decimal to Binary.

Step 1 : Multiply the decimal fraction by 2 and note the integer part. The integer part is either 0 or 1.

Step 2 : Discard the integer part of the previous product. Multiply the fractional part of the previous product by 2. Repeat step 1 until the same fraction repeats or terminates (0).

Step 3 : The resulting integer part forms a sequence of 0's and 1's that becomes the binary equivalent of decimal fraction.

Step 4 : The final answer is to be written from first integer part obtained till the last integer part obtained.

(b) Convert $(98.46)_{10}$ to Binary

Binary Equivalent of $(98)_{10} \rightarrow (?)_2$

2	98
2	49 - 0
2	24 - 1
2	12 - 0
2	6 - 0
2	3 - 0
2	1 - 1

$(98)_{10} = (1100010)_2$

Fractional Part: $(.46)_{10} \rightarrow (?)_2$

$$\begin{aligned} 0.46 \times 2 &= 0.92 \\ 0.92 \times 2 &= 1.84 \\ 0.84 \times 2 &= 1.68 \\ 0.68 \times 2 &= 1.36 \\ 0.36 \times 2 &= 0.72 \end{aligned}$$

$(.46)_{10} = (01110)_2$

$(98.46)_{10} = (1100010.01110)_2$

2. Find 1's Complement and 2's Complement for the following Decimal number a) -98

b) -135

a) -98

1's complement

2	98
2	49 - 0
2	24 - 1
2	12 - 0
2	6 - 0
2	3 - 0
2	1 - 1

$(98)_{10} = (01110010)_2$

1's complement = 1 0 0 1 1 1 0 1

2's complement \Rightarrow 1 0 0 1 1 1 1 0

$\therefore (98)_{10} = (10011110)_2$

b) -135

2's complement

2	135
2	67 - 1
2	33 - 1
2	16 - 1
2	8 - 0
2	4 - 0
2	2 - 0
2	1 - 0

$(135)_{10} = 10000111$

1's complement = 0 1 1 1 1 0 0 0

2's complement = 0 1 1 1 1 0 0 1

$\therefore (135)_{10} = (01111001)_2$

3. a) Add: $1101010_2 + 101101_2$

b) Subtract: $1101011_2 - 111010_2$

a) Add $1101010_2 + 101101_2$

$$\begin{array}{r} 01101010 \\ 00101101 \\ \hline 10010111_2 \end{array}$$

b) Subtract $1101011_2 - 111010_2$

$$\begin{array}{r} 01101011 \\ 00111010 \\ \hline 00110001_2 \end{array}$$

BOOLEAN ALGEBRA

Section - B

1. What is Boolean Algebra?

- Boolean algebra is a mathematical discipline that is used for designing digital circuits in a digital computer. It describes the relation between inputs and outputs of a digital circuit.

2. Write a short note on NAND Gate.

- The NAND gate is the combination of NOT and AND gates. The NAND gate is generated by inverting the output of a AND gate. The algebraic expression of the NAND is $Y = \overline{A.B}$

3. Draw the truth table for XOR gate.

Input		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

4. Write the associative laws?

$A + (B + C) = (A + B) + C$

$A.(B.C) = (A.B).C$

5. What are derived gates?

- Derived Gates are the gates which are derived from the fundamental gates.
- Example : NAND, NOR, XOR, XNOR

Section - C

1. Write the truth table of fundamental gates.

(a) AND gate – Truth Table

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

(b) OR gate – Truth Table

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

(c) NOT Gate – Truth Table

A	\bar{A}
0	1
1	0

2. Write a short note on XNOR gate.

- The XNOR (exclusive – NOR) gate is a combination of XOR gate followed by an inverter.
- Its output is “true” if the inputs are same and false if the inputs are different.

The output of XNOR is $C = A \odot B$

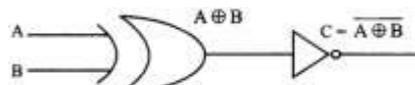
Truth Table:

Input		Output
A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

Logic symbol:



Logic circuit:



3. Reason out why the NAND and NOR are called universal gates?

- NAND and NOR gates are called as Universal gates because the fundamental logic gates can be realized through them.

4. Give the truth table of XOR gate.

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

5. Write the De Morgan's law.

$$\overline{(A + B)} = \bar{A} . \bar{B}$$

$$\overline{(A . B)} = \bar{A} + \bar{B}$$

Section - D

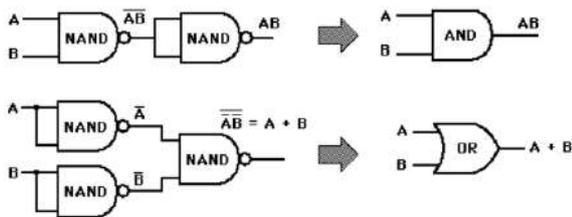
1. Explain the fundamental gates with expression and truth table.

- A gate is a basic electronic circuit which operates on one or more signals to produce an output signal.
- The three fundamental gates are AND, OR and NOT gates.

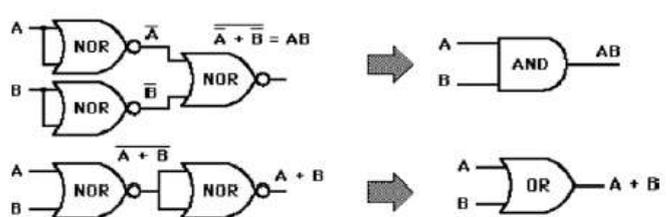
Gates	Description	Truth Table																		
AND 	The AND gate can have two or more input signals and produce an output signal. The output is “true” only when both inputs are “true”, otherwise, the output is “false”.	<table border="1"> <thead> <tr> <th colspan="2">Input</th> <th>Output</th> </tr> <tr> <th>A</th> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Input		Output	A	B	C	0	0	0	0	1	0	1	0	0	1	1	1
Input		Output																		
A	B	C																		
0	0	0																		
0	1	0																		
1	0	0																		
1	1	1																		
OR 	The OR gate gets its name from its behaviour like the logical inclusive “OR”. The output is “true” if either or both of the inputs are “true”. If both inputs are “false” then the output is “false”.	<table border="1"> <thead> <tr> <th colspan="2">Input</th> <th>Output</th> </tr> <tr> <th>A</th> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Input		Output	A	B	C	0	0	0	0	1	1	1	0	1	1	1	1
Input		Output																		
A	B	C																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
NOT $C = \bar{A}$ 	The NOT gate, called a logical inverter, has only one input. It reverses the logical state. In other words the output C is always the complement of the input.	<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> <tr> <th>A</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </tbody> </table>	Input	Output	A	C	1	0	0	1										
Input	Output																			
A	C																			
1	0																			
0	1																			

2. How AND and OR can be realized using NAND and NOR gate.

i) Realization of OR using NAND:



ii) Realization of AND using NOR:



3. Explain the Derived gates with expression and truth table.

The logic gates like NAND, NOR, XOR and XNOR are derived gates which are derived from the fundamental gates AND, OR NOT.

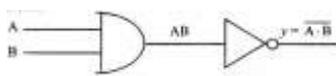
(i) NAND gate:

The NAND is the combination of NOT and AND. The NAND is generated by inverting the output of an AND operation.

Logic circuit

Logic symbol:

Truth table:



Inputs		Outputs
A	B	$y = \overline{A \cdot B}$
0	0	1
0	1	0
1	0	0
1	1	0

Inputs Outputs

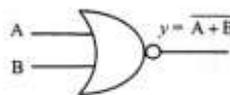
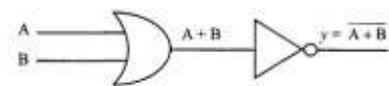
(ii) NOR gate:

The NOR is the combination of NOT and OR. The NOR is generated by inverting the output of an OR operation.

Logic circuit

Logic symbol:

Truth table:



Inputs		Outputs
A	B	$y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

CHAPTER-3

COMPUTER ORGANISATION

SECTION – B : Short Answers

1. What are the parameters which influence the characteristics of a microprocessor?

- The parameters which influence the characteristics of a microprocessor are
 - Clock speed
 - Instruction set
 - Word size

2. What is an instruction?

- Command which is given to the computer to do an operation on data is called an instruction,

3. What is a program counter?

- A program counter (PC) is a special register which always keeps the address of a next instruction to be executed.

4. What is HDMI?

- HDMI (High Definition Multimedia Interface)
- It is an audio/video interface which transfer the uncompressed video and audio data from a video controller, to a computer monitor, LCD projector, digital television, etc.

5. Which source is used to erase the content of a EPROM?

- Ultra violet rays are used to erase the contents of EPROM.

SECTION – C

III. Explain in Brief

1. Differentiate Computer Organisation from Computer Architecture.

Computer Organization	Computer Architecture
It deals with the hardware components	It deals with structure and behaviour of computer system
It describes how the computer performs	It describes what the computer does

2. Classify the microprocessor based on the size of the data.

- Microprocessor can be classified as follows
 - 8 bit microprocessor
 - 16 bit microprocessor
 - 32 bit microprocessor
 - 64 bit microprocessor.

3. Write down the classifications of microprocessors based on the instruction set.

- Classification of Microprocessor based on the Instruction set:
 - RISC- Reduced Instruction Set Computers.
 - CISC- Complex Instruction set Computers.

4. Differentiate PROM and EPROM.

PROM	EPROM
Programmable Read Only Memory	Erasable Programmable Read Only Memory
It can be written but cannot be erased	It can be erased and reprogrammable
Inexpensive	Expensive

5. Write down the interfaces and ports available in a computer.

- Serial Port
- Parallel Port
- USB Ports
- USB 3
- HDMI
- VGA Connector
- Audio Plugs
- PS/2 Port
- SCSI Port

6. Differentiate CD and DVD.

CD	DVD
Compact Disc	Digital Versatile Disc
Store upto 700 MB of data	Store upto 4.7 GB of data
It cannot play DVD's	It can play CD's

7. How will you differentiate a flash memory and an EEPROM?

Flash Memory	EEPROM
Faster in performance	Slower in performance
Storage capacity 1 GB to 100 GB	Storage capacity 1 KB to 2 MB
Access and erase the data blockwise	Access and erase the data byte-wise

SECTION – D: Explain in Detail

1. Explain the characteristics of a microprocessor.

- Characteristics of a microprocessor are,
 - (i) Clock speed
 - (ii) Instructions set
 - (iii) Word size

(i) Clock speed:

- Each microprocessor has an internal clock that regulates the speed at which it executes instructions.
- The speed at which the microprocessor executes instructions is called clock speed.
- It is measured in MHz or in GHz.

(ii) Instructions set:

- The set of machine level instructions that a microprocessor is designed to execute is called as an instruction set.
- This carries following types of operations,
 - Data transfer
 - Arithmetic operations
 - Logical Operations
 - Control flow
 - Input/output

(iii) Word size:

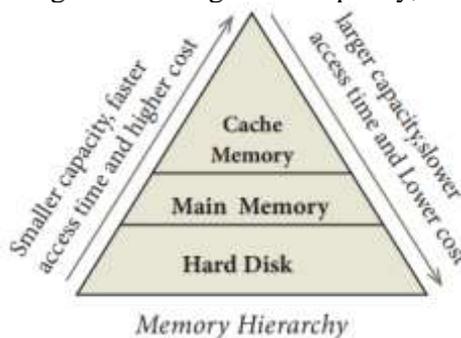
- The number of bits that can be processed by a processor in a single instruction is called as its word size.
- It determines the amount of RAM that can be accessed at one time and the total number of pins on the microprocessor.
- Total number of input and output pins in turn determines the architecture of the microprocessor.

2. How the read and write operations are performed by a processor? Explain.

- The Central Processing Unit(CPU) has a Memory Data Register (MDR) and a Memory Address Register (MAR).
- The Memory Data Register (MDR) keeps the data which is transferred between the Memory and the CPU. The Program Counter (PC) is a special register in the CPU which always keeps the address of the next instruction to be executed.
- The Arithmetic and Logic unit of CPU places the address of the memory to be fetched, into the Memory Address Register.
- A bus is a collection of wires used for communication between the internal components of a computer.
- The read operation transfers the data(bits) from word to Memory Data Register.
- The write operation transfers the data(bits) from Memory Data Register to word.

3. Arrange the memory devices in ascending order based on the access time.

- Different memory devices are arranged according to the capacity,



4. Explain the types of ROM.

Types of ROM

i) Read Only Memory (ROM):

- A computer with pre-recorded data at manufacturing time which cannot be modified.
- It stores critical programs that boots the computer.
- It can only be read.
- It can retain its contents even when the computer is turned off.
- It is called as a non – volatile memory.

ii) Programmable Read Only Memory (PROM):

- It is also a non – volatile memory.
- It can retain their contents even when the computer is turned off.
- PROM is manufactured as a blank memory.
- A PROM burner is used to write data to a PROM chip. It is called burning the PROM.

iii) Erasable Programmable Read Only Memory (EPROM):

- It is a special type of memory which serves as a PROM, but the content can be erased using ultraviolet rays.
- The ultraviolet light clears its contents, making it possible to reprogram the memory.
- EPROMs are used widely in personal computers

iv) Electrically Erasable Programmable Read Only Memory (EEPROM):

- It is a special type of PROM that can be erased by exposing it to an electrical charge.
- It can retain its contents even when the power is turned off.
- It is slower in performance.

CHAPTER - 4 THEORETICAL CONCEPTS OF OPERATING SYSTEM

SECTION-B : Very Short Answers

1) List out any two uses of Operating System?

- Easy interaction between the users and computers
- Starting computer operation automatically when the power is turned on (Booting)
- Providing security to user programs
- Controlling Input and Output Devices

2) What is multi-user Operating system?

- It is used in computers and laptops that allow same data and applications to be accessed by multiple users at the same time.
- The users can also communicate with each other.
- Windows, Linux and UNIX are examples for multi-user Operating System.

3) What is a GUI?

- The GUI is a window based system with a pointing device to direct I/O, choose from menus, make selections and a keyboard to enter text.

4) What are the security management features available in Operating System?

- File access level
- System level
- Network level

5) What is multi-processing?

- It has two or more processors for a single running process (job).
- Processing takes place in parallel is known as parallel processing.
- Since the execution takes place in parallel, this feature is used for high speed execution which increases the power of computing.

6) What are the different Operating Systems used in computer?

- Windows, UNIX and Linux

SECTION-C : Short Answers

1) What are the advantages and disadvantages of Time-sharing features?

Advantages

- Every task gets an equal
- Less chances of duplication of software
- CPU idle time can be reduced

Disadvantages

- Reliability problem
- One must have to take care of security and integrity of user programs and data
- Data communication problem

2) List out the key features of Operating system

- User Interface
- File Management
- Memory Management
- Process Management
- Security Management
- Fault Tolerance

3) Write a note on Multiprocessing

- It has two or more processors for a single running process (job).
- Processing takes place in parallel is known as parallel processing.
- Since the execution takes place in parallel, this feature is used for high speed execution which increases the power of computing.

SECTION - D : Explain in detail

1) Explain the concept of a Distributed Operating System along with its advantages.

- The Distributed Operating System is used to access shared data and files that reside in any machine around the world using internet/intranet.
- The users can access as if it is available on their own computer.

Advantages

- A user at one location can make use of all the resources available at another location over the network.
- Many computer resources can be added easily in the network
- Improves the interaction with the customers and clients.
- Reduces the load on the host computer.

2) List out the points to be noted while creating a user interface for an Operating system.

- The GUI is a window based system with a pointing device to direct I/O, choose from menus, make selections and a keyboard to enter text.

- **Points to be noted while creating a user interface for an Operating system**
 - The user interface should enable the user to retain this expertise for a longer time.
 - The user interface should also satisfy the customer based on their needs.
 - The user interface should save user's precious time.
 - The ultimate aim of any product is to satisfy the customer. The User Interface is also to satisfy the customer.
 - The user interface should reduce number of errors committed by the user

3) Explain the process management algorithms in Operating System.

- **FIFO (First In First Out)Scheduling**
 - This algorithm is based on queuing technique. the process that enters the queue first is executed first by the CPU, followed by the next and so on. The processes are executed in the order of the queue (row).
- **SJF (Shortest Job First)Scheduling:**
 - This algorithm works based on the size of the job being executed by the CPU.
- **Round Robin Scheduling**
 - The Round Robin (RR) scheduling algorithm is designed especially for time sharing systems. Jobs (processes) are assigned and processor time in a circular method.
- **Based On Priority**
 - The given job (process) is assigned based on a Priority. The job which has higher priority is more important than other jobs.

CHAPTER-5 WORKING WITH WINDOWS OPERATING SYSTEM

SECTION-B : Very Short Answers

1. What is known as Multitasking?

- Multiple applications can execute simultaneously in Windows, and this is known as "Multitasking".

2. What are called standard icons

- The icons which are available on desktop by default while installing Windows OS are called standard icons.
- The standard icons available in all Windows OS are My Computer, Documents and Recycle Bin.

3. Differentiate Files and Folders.

Files	Folders
Files contain data	It store files and folders
It is the collection of records	It is the collection of files

4. Differentiate Save and save As option.

Save	Save As
It save a document	This option save an already saved document
This command opens a dialog box, only when the file saved for the first time	This Save as command opens a dialog box every time

5. How will you Rename a File?

- There are number of ways to rename files or folders. They are,
 - Using the File menu.
 - Using the left mouse button.
 - Using the right mouse button.

SECTION-C : Short Answers

1. What are the functions of Windows Operating system.

- Access applications (programs) on the computer (word processing, games, spread sheets, calculators and so on).
- Load any new program on the computer.
- Manage hardware such as printers, scanners, mouse, digital cameras etc.,
- File management activities (For example creating, modifying, saving, deleting files and folders).
- Change computer settings such as colour scheme, screen savers of your monitor, etc

2. Write a note on Recycle bin.

- Recycle bin is a special folder to keep the files or folders deleted by the user, which means you still have an opportunity to recover them. The user cannot access the files or folders available in the Recycle bin without restoring it.
- To restore file or folder from the Recycle Bin

- Open Recycle bin.
- Right click on a file or folder to be restored and select Restore option from the pop-up menu.
- To restore multiple files or folders, select Restore all items.
- To delete all files in the Recycle bin, select Empty the Recycle Bin.

3. Write a note on the elements of a window.

Title Bar

- The title bar will display the name of the application and the name of the document opened. It will also contain minimize, maximize and close button.

Menu Bar

- The menu bar is seen under the title bar.
- Menus in the menu bar can be accessed by pressing Alt key and the letter that appears underlined in the menu title.
- Additionally, pressing Alt or F10 brings the focus on the first menu of the menu bar.

The Workspace

- The workspace is the area in the document window to enter or type the text of your document.

Scroll bars

- The scroll bars are used to scroll the workspace horizontally or vertically.

Corners and borders

- The corners and borders of the window helps to drag and resize the windows. The mouse pointer changes to a double headed arrow when positioned over a border or a corner.

4. Write the two ways to create a new folder.

- There are two ways in which you can create a new folder:

- **Method I:**

- Step 1: Open Computer Icon.
- Step 2: Open any drive where you want to create a new folder.
- Step 3: Click on File → New → Folder.
- Step 4: A new folder is created with the default name “New folder”.
- Step 5: Type in the folder name and press Enter key.

- **Method II:**

- In order to create a folder in the desktop:

- Step 1: In the Desktop, right click → New → Folder.
- Step 2: A Folder appears with the default name “New folder” and it will be highlighted
- Step 3: Type the name you want and press Enter Key.
- Step 4: The name of the folder will change.

5. Differentiate Copy and Move

Copy	Move
Copying files and folders	Moving files and folders
Do not remove files or folders	Remove files or folders
Command: Edit → Copy (or) Ctrl + C Edit → Paste (or) Ctrl + V	Command: Edit → Cut (or) Ctrl + X Edit → Paste (or) Ctrl + V

SECTION - D : Explain in detail

1. Explain the versions of Windows Operating System.

Version	Year	Specific features
Windows 1.x	1985	<ul style="list-style-type: none"> • Introduction of GUI in 16 - bit. processor • Mouse was introduced as an input device.
Windows 2.x	1987	<ul style="list-style-type: none"> • Supports to minimize or maximize windows. • Control panel feature was introduced with various system settings and customising options.
Windows 3.x	1992	<ul style="list-style-type: none"> • Introduced the concept of multitasking. • Supported 256 colours which brought a more modern, colourful look to the interface.
Windows 95	1995	<ul style="list-style-type: none"> • Introduced Start button, the taskbar, Windows Explorer and Start menu. • Introduced 32 - bit processor and focused more on multitasking
Windows 98	1998	<ul style="list-style-type: none"> • Integration of the Web browser (Internet Explorer) with the Operating

		System. <ul style="list-style-type: none"> • DOS gaming began to disappear as Windows based games improved. • Plug and play feature was introduced.
Windows NT		<ul style="list-style-type: none"> • Designed to act as servers in network.
Windows Me	2000	<ul style="list-style-type: none"> • It introduced automated system diagnostics and recovery tools.
Windows 2000	2000	<ul style="list-style-type: none"> • Served as an Operating System for business desktop and laptop systems. • Four versions of Windows 2000 were released: <ul style="list-style-type: none"> ○ Professional, ○ Server, ○ Advanced Server and ○ Data Centre Server.
Windows XP	2001	<ul style="list-style-type: none"> • Introduced 64-bit Processor. • Improved Windows appearance with themes and offered a stable version.
Windows Vista	2006	<ul style="list-style-type: none"> • Updated the look and feel of Windows.
Windows 7	2009	<ul style="list-style-type: none"> • Booting time was improved, introduced new user interfaces like Aero Peek, pinning programs to taskbar, handwriting recognition etc. and Internet Explorer 8.
Windows 8	2012	<ul style="list-style-type: none"> • Windows 8 is faster than previous versions of Windows. • Start button was removed. • Windows 8 takes better advantage of multi-core processing, solid state drives (SSD), touch screens and other alternate input methods. • Served as common platform for mobile and computer
Windows 10	2015	<ul style="list-style-type: none"> • Start Button was added again. • Multiple desktop. • Central Notification Center for App notification and quick actions. • Cortana voice activated personal assistant

2. Explain the different ways of finding a file or Folder

Finding Files and Folders

- User can use the search box on the Start menu to quickly search a particular folder or file in the computer or in a specific drive.

To find a file or folder:

1. Click the Start button, the search box appears at the bottom of the start menu.
2. Type the name of the file or the folder you want to search. Even if you give the part of the file or folder name, it will display the list of files or folders starting with the specified name. 3. The files or the folders with the specified names will appear, if you click that file, it will directly open that file or the folder.
3. There is another option called “See more results” which appears above the search box.
4. If you click it, it will lead you to a Search Results dialog box where you can click and open that file or the folder.

Searching Files or folders using Computer icon

1. Click Computer Icon from desktop or from Start menu.
2. The Computer disk drive screen will appear and at the top right corner of that screen, there is a search box option.
3. Type the name of the file or the folder you want to search. Even if you give the part of the file or folder name, it will display the list of files or folders starting with the specified name
4. Just click and open that file or the folder.

3. Write the procedure to create shortcut in Windows OS.

- Select the file or folder that you wish to have as a shortcut on the Desktop.
- Right click on the file or folder.
- Select Send to → Desktop (create shortcut) from the sub-menu.
- A shortcut for the file or folder will now appear on your desktop and you can open it from the desktop in the same way as any other icon.

CHAPTER - 6 : SPECIFICATION AND ABSTRACTION

Section - B : Answer the following questions

1. Define an algorithm.

- An algorithm is a step-by-step sequence of statements to solve a problem

2. Distinguish between an algorithm and a process.

Algorithm	Process
It is a sequence of statements to solve a problem	When the statements are executed, a process workout and solves the given problem
It has rules to execute Example: Recipe to bake a cake.	It is an action to achieve a result Example: Baking of cake (ie) cooking

3. Initially, farmer, goat, grass, wolf L, L, L, L and the farmer crosses the river with goat.

Model the action with an assignment statement.

Farmer, goat := R, R

4. Specify a function to find the minimum of two numbers.

Solution:

1. minimum (x,y,z)
2. - - inputs: x ,y and z are integer number (or) real numbers
3. - - outputs: $x < y$ and $x < z$ x is minimum,
 $y < x$ and $y < z$ y is minimum,
 $z < x$ and $z < x$ z is minimum.

5. If $\sqrt{2}=1.414$, and the square root() function returns -1.414, does it violate the following specification.

- - square root (x)

- - inputs: x is a real number, $x \geq 0$

- - outputs: y is a real number such that $y^2=x$

Solution:

- - square-root

- - input: x is a real number, $x > 0$

- - output: y is a real number such that $y = \sqrt{x}$

Yes, it violate the specification. The square root of a number is a positive number. Hence the output will not be negative.

Section - C : Answer the following questions in brief

1. When do you say that a problem is algorithmic in nature?

We can say that a problem is algorithmic in nature:

- (i) When its solution involves the construction of an algorithm
- (ii) Some types of problems can be immediately recognized as algorithmic.

2. What is the format of the specification of an algorithm?

Let X be the required input

Y be the desired output, then the algorithm is specified as:

1. Name of an algorithm (inputs)
2. - - inputs: X
3. - - outputs: Y

3. What is abstraction?

- Ignoring or hiding unnecessary details and modeling a problem only by its essential features is known as abstraction.

4. How is state represented in algorithms?

- State can be represented by a set of variables in an algorithm.
- The state at any point is simply the values of the variables at the point.

5. What is the form and meaning of assignment statement?

Form: Variable:= expression (or) value.

Assignment Statement:

- It is used to store a value in a variable.
- It is written with the variable on the left side of assignment operator and value on the right side
variable := value.
- When the assignment is executed, the value on the right side is stored in the variable on the left side.

6. What is the difference between assignment operator and equality operator?

Assignment operator	Equality operator
'=' is the assignment operator	'== ' is the equality operator
It is used to assign the result of the expression on the right side to variable on the left side	It is used to check whether the two expression are equal on the both sides or not

Section - D

Explain the following questions in detail

1. Write the specification of an algorithm hypotenuse whose inputs are the lengths of the two shorter sides of a right angled triangle, and the output is the length of the third side.

Answer:

1. hypotenuse (a, b)
2. -- inputs: a, b are real numbers, $a > 0, b > 0$
3. -- outputs: $c^2 = a^2 + b^2$ where c is real number, $c > 0$

2. Suppose you want to solve the quadratic equation $ax^2 + bx + c = 0$ by an algorithm. quadratic_solve (a, b, c)

- inputs : ?
- outputs: ?

You intend to use the formula and you are prepared to handle only real number roots. Write a suitable specification.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Answer:

1. Quadratic – solve (a, b, c)
2. -- inputs: a, b, c are real numbers, $a \neq 0$
3. -- outputs: x is a real number, such that $ax^2 + bx + c = 0$ is satisfied by two values

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

3. Exchange the contents: Given two glasses marked A and B. Glass A is full of apple drink and glass B is full of grape drink. For exchanging the contents of glasses A and B, represent the state by suitable variables, and write the specification of the algorithm.

Answer:

Let the variables a, b, c represent the glass A, glass B and Glass C respectively. Variables A, B, C can store values APPLE, GRAPE or EMPTY.

- Initial State:
1. -- a, b, c := APPLE, GRAPE, EMPTY
 2. c := b
 3. -- a, b, c := APPLE, EMPTY, GRAPE
 4. b := a
 5. -- a, b, c := EMPTY, APPLE, GRAPE
 6. a := c
- Final State:
7. -- a, b, c := GRAPE, APPLE, EMPTY

Specification:

1. Exchange ()
2. -- inputs: a, b := APPLE, GRAPE
3. -- outputs: a, b := GRAPE, APPLE

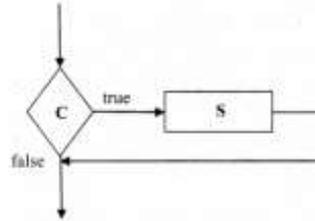
CHAPTER - 7 : COMPOSITION AND DECOMPOSITION

Section - B : Answer the following questions in one or two lines

1. Distinguish between a condition and a statement.

Condition	Statement
It is a phrase that describes a test of the state	It is a phrase that commands the computer to do an action

2. Draw a flowchart for conditional statement.



3. Both conditional statement and iterative statement have a condition and a statement. How do they differ?

Conditional statements	Iterative Statements
It is executed only if the condition is true otherwise, nothing is done	It evaluation repeatedly as long as the condition is true

4. What is the difference between an algorithm and a program?

Algorithm	Program
Step-by-step operations are performed to solve the problem	Sequence of instructions given to the computer using specific programming language.

5. Why is function an abstraction?

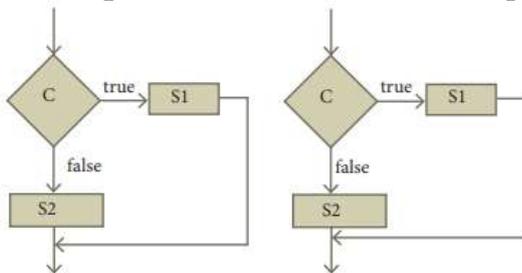
- A function is an abstraction of a sub-problems and it is specified by the input property and its input-output relation.

6. How do we refine a statement?

- Each sub problem is expanded into more detailed statements (or) finer steps in the subsequent level.

Section - C : Answer the following questions in brief

1. For the given two flowcharts write the pseudo code.



Answer : i)	if C
	S1
	else
	S2
ii)	if C
	S

2. If C is false in line 2, trace the control flow in this algorithm.

1. S1
2. -- C is false
3. if C
4. S2
5. else
6. S3
7. S4

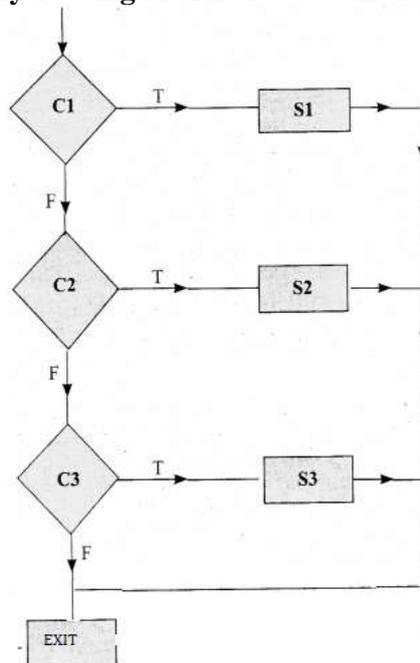
Answer: S1 , S3, S4

3. What is case analysis?

- Case analysis splits the problem into exhaustive set of disjoint cases. For each case, the problem is solved independently.

1. Case C1
 2. S1
 3. Case C2
 4. S2
 5. Case C3
 6. S3
 7. else
- S4

4. Draw a flowchart for -3 case analysis using alternative statements.



5. Define a function to double a number in two different ways:

1. $n + n$
2. $2 \times n$

Answer:

1. Double (n)

-- inputs: n is a real number or an integer, $n > 0$

-- Outputs: y is a real number or an integer such that $y = n + n$

2. Double (n)

-- inputs: n is a real number or an integer, $n > 0$

-- Outputs: y is a real number or an integer such that $y = 2 \times n$

Section - D : Explain the following questions in detail

1. Exchange the contents: Given two glasses marked A and B. Glass A is full of apple drink and glass B is full of grape drink. Write the specification for exchanging the contents of glasses A and B, and write a sequence of assignments to satisfy the specification.

Answer:

- Let the variables a, b, c represent Glass A, Glass B and Glass C and a, b, c can store values APPLE, GRAPE or EMPTY.

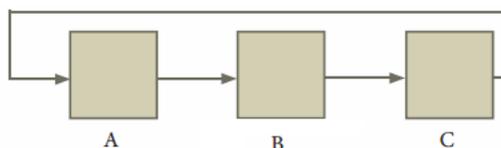
Specification:

1. exchange (a, b)
2. -- inputs: a, b := APPLE, GRAPE
3. -- outputs: a, b := GRAPE, APPLE

Algorithm:

1. exchange (a, b)
2. -- a, b, c := APPLE, GRAPE, EMPTY.
3. c := b
4. -- a, b, c := APPLE, EMPTY, GRAPE,
5. b := a
6. -- a, b, c := EMPTY, APPLE, GRAPE
7. a := c
8. -- a, b, c := GRAPE, APPLE, EMPTY

2. Circulate the contents: Write the specification and construct an algorithm to circulate the contents of the variables A, B and C as shown below: The arrows indicate that B gets the value of A, C gets the value of B and A gets the value of C.



Specifications:

1. circulate
2. -- inputs: a, b, c: = A, B, C
3. -- outputs: a, b, c: = C, A, B

Algorithm:

1. circulate (a, b, c)
2. -- a, b, c: = A, B, C
3. temp : = c
4. c : = b
5. b : = a
6. a : = temp
7. -- a, b, c: = C, A, B

3. Decanting problem. You are given three bottles of capacities 5, 8, and 3 litres. The 8L bottle is filled with oil, while the other two are empty. Divide the oil in 8L bottle into two equal quantities. Represent the state of the process by appropriate variables. What are the initial and final states of the process? Model the decanting of oil from one bottle to another by assignment. Write a sequence of assignments to achieve the final state.

Answer:

- Let a, b, c be the variables whose maximum value are 8L, 5L and 3L respectively.

Initial State: a, b, c: = 8, 0, 0

Final State: a, b, c: = 4, 4, 0

Specifications:

1. decant
2. -- inputs: a, b, c : = 8, 0, 0
3. -- outputs: a, b, c : = 4, 4, 0

Algorithm:

Let us assume that a: = b denote oil in b is poured into a bottle until either “a” is full or “b” becomes empty.

1. decant (a, b, c)
2. -- a, b, c : = 8, 0, 0
3. b : = a
4. -- a, b, c : = 3, 5, 0
5. c : = b
6. -- a, b, c : = 3, 2, 3
7. a : = c
8. -- a, b, c : = 6, 2, 0
9. c : = b
10. -- a, b, c : = 6, 0, 2
11. b : = a
12. -- a, b, c : = 1, 5, 2
13. c : = b
14. -- a, b, c : = 1, 4, 3
15. a : = c
16. -- a, b, c : = 4, 4, 0

4. Trace the step – by – step execution of the algorithm for factorial(4).

factorial(n)

-- inputs : n is an integer , n ≥ 0

-- outputs : f = n!

f, i : = 1, 1

while i ≤ n

f, i : = f x i, i + 1

Answer :

(i)	f = 1	i = 1	f = f * i	i = i + 1
			f = 1 × 1 = 1	i = 2
(ii)			f = 1 × 2	i = 3
			= 2	
(iii)			f = 2 × 3	i = 4
			= 6	
(iv)			f = 6 × 4	i = 5 (loop terminates)
			= 24	

CHAPTER-8 : ITERATION AND RECURSION

SECTION-B : Very Short Answers

1. What is an invariant?

- Property of the variables which remains unchanged by the execution of the body is called an invariant.

2. Define a loop invariant.

- The unchanged variables by the execution of the loop body is called the loop invariant.

3. Does testing the loop condition affect the loop invariant? Why?

- Yes, it affects.
- The loop body is repeatedly executed as long as the loop conditions is true. Each time the loop body is executed, the variables are updated.

4. What is the relationship between loop invariant, loop condition and the input- output recursively

- Loop invariant is a condition that is necessarily true before the loop body and after the body each time
- When the loop ends, the termination condition and loop invariant should establish the input, output relation

5. What is recursive problem solving?

- Recursion steps breaks the problem into sub-problems of smaller size, assume solutions for sub-problems are given by the recursive calls, and construct solutions to the given problem.

6. Define factorial of a natural number recursively.

```

Factorial (n)
    -- inputs: n
    -- outputs: factorial of n
    if n = 0 -- base case
        1
    else
        n * factorial (n - 1) -- recursion step
    
```

SECTION-C : Short Answers

1. There are 7 tumblers on a table, all standing upside down. You are allowed to turn any 2 tumblers simultaneously in one move. Is it possible to reach a situation when all the tumblers are right side up? (Hint: The parity of the number of upside down tumblers is invariant.)

Answer:

Let

u – No. of tumblers right side up

v – No. of tumblers up side down

Initial stage : u = 0, v = 7 (All tumblers upside down)

Final stage output: u = 7, v = 0 (All tumblers right side up)

Possible Iterations:

(i) Turning both up side down tumblers to right side up

$$u = u + 2, v = v - 2 \text{ [u is even]}$$

(ii) Turning both right side up tumblers to upside down.

$$u = u - 2, v = v + 2 \text{ [u is even]}$$

(iii) Turning one right side up tumblers to upside down and other tumbler from upside down to right side up.

$$u = u + 1 - 1 = u, v = v + 1 - 1 = v \text{ [u is even]}$$

Initially u = 0 and continuous to be even in all the three cases. Therefore u is always even. Invariant: u is even

(i. e. No. of right side up tumblers are always even)

But in the final stage (Goal), u = 7 and v = 0 i. e. u is odd.

Therefore it is not possible to reach a situation where all the tumblers are right side up.

2. A knockout tournament is a series of games. Two players compete in each game; the loser is knocked out (i.e. does not play any more), the winner carries on. The winner of the tournament is the player that is left after all other players have been knocked out. Suppose there are 1234 players in a tournament. How many games are played before the tournament winner is decided?

Answer:

No. of players	2	3	4	5	n	1234
No. of games	1	2	3	4	n - 1	1234 - 1 = 1233

Let

r → no. of players remaining

n → no. of games played

After every game, r will be reduced by 1.

If $r = 2$ then $n = 1$

As n increases, r decreases

$$n, r := n + 1, r - 1$$

$$n + r = (n + 1) + (r - 1)$$

$$= n + 1 + r - 1$$

$$= n + r$$

Therefore $n + r$ is invariant.

$n + r = 1234$ (No. of players initially)

After all the games, only one player (winner) is left out. i. e. $n = 1$

$$n + r = 1234$$

$$1 + r = 1234$$

$$r = 1234 - 1 = 1233$$

No. of games played = 1233

3. King Vikramaditya has two magic swords. With one, he can cut off 19 heads of a dragon, but after that the dragon grows 13 heads. With the other sword, he can cut off 7 heads, but 22 new heads grow. If all heads are cut off, the dragon dies. If the dragon has originally 1000 heads, can it ever die? (Hint: The number of heads mod 3 is invariant.)

Answer:

No. of heads of dragon = 1000

sword 1 : cuts 19 heads but 13 heads grow back.

sword 2 : cuts 7 heads but 22 heads grow back.

Let n be the number of heads of the dragon at initial state.

Case 1 : King uses Sword 1

Sword 1 cuts off 19 heads but 13 heads grow back.

$$n := n - 19 + 13 = n - 6$$

No. of heads are reduced by 6.

Case 2 : King uses Sword 2

Sword 2 cuts 7 heads but 22 heads grow back.

$$n := n - 7 + 22 = n + 15$$

No. of heads are increased by 15.

- In the above two cases either 6 heads are removed or 15 heads added.
- Both 6 and 15 are multiples of 3. Therefore repeating case 1 and case 2 recursively, will either reduce or increase dragon heads in multiples of 3.
- That is the invariant is $n \bmod 3$.
- If $n \bmod 3 = 0$ then there is a possibility that the dragon dies.
- But 1000 is not a multiple of 3
 - $1000 \bmod 3 = 1 \neq 0$
- It is not possible to kill the dragon. The dragon never dies.

SECTION - D

Explain in detail

1. Assume an 8×8 chessboard with the usual coloring. "Recoloring" operation changes the color of all squares of a row or a column. You can recolor repeatedly. The goal is to attain just one black square. Show that you cannot achieve the goal. (Hint: If a row or column has b black squares, it changes by $(8 - b) - b$).

Answer

Let B be Black square and W be White square

Consider that the chessboard starts with white square.

i. Initially there will be 4 white and 4 black i.e. $B=4$ and $W=4$

ii. If Change the first square from white to black then $B=5$ and $W=3$

iii. If Change the second square from black to white then $B=4$ and $W=4$

iv. If we continue the step-i and step-ii, at the end of each row, we will $B=4$ and $W=4$.

So, it is not possible to attain the goal just one black square.

2. Power can also be defined recursively as

$$a^n = \begin{cases} 1 & \text{if } n = 0 \\ a \times a^{n-1} & \text{if } n \text{ is odd} \\ a^{n/2} \times a^{n/2} & \text{if } n \text{ is even} \end{cases}$$

Construct a recursive algorithm using this definition. How many multiplications are needed to calculate a^{10} ?

Answer:

```

power (a, n)
-- inputs : a , n are positive integers, n ≥ 0
-- outputs: an
if n = 0
    1
else
    if n mod 2 = 1
        a x power (a , n - 1)
    else
        power( a, n/2) x power (a, n/2)

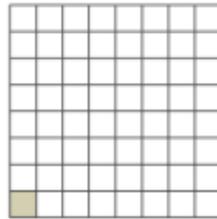
```

3. A single-square-covered board is a board of $2^n \times 2^n$ squares in which one square is covered with a single square tile. Show that it is possible to cover the this board with triominoes without overlap.

Answer:

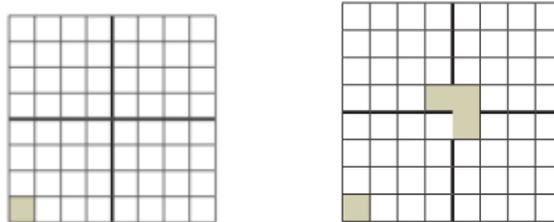
We can solve the problem by recursion.

The base case is $n = 1$. It is a 2×2 corner-covered board.



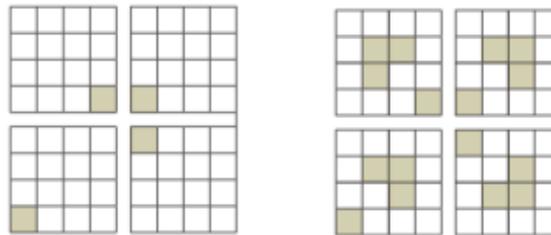
In the recursion step, divide the corner covered board of size $2^n \times 2^n$ into 4 sub-boards, each of size $2^{n-1} \times 2^{n-1}$, by drawing horizontal and vertical lines through the centre of the board.

Place a triominoe at the center of the entire board so as to not cover the corner-covered sub-board



We have 4 sub-problems whose size is strictly smaller than the size of the given problem.

We can solve each of the sub-problems recursively.

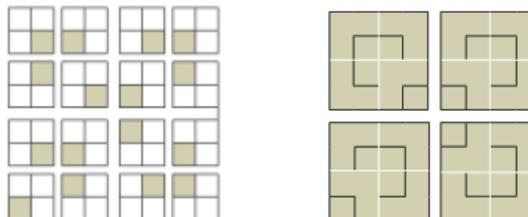


Tile corner_covered board of size n

```

if n = 1 -- base case
    cover the 3 squares with one triominoe
else -- recursion step
    divide board into 4 sub_boards of size n-1
    place a triominoe at centre of board, leaving out the corner_covered sub-board tile each
    sub_board of size n-1

```



CHAPTER - 9 PART-I : INTRODUCTION TO C++

SECTION-B : Very Short Answers

1. What is meant by a token? Name the token available in C++.

- The smallest individual unit in a program is known as token or lexical unit or lexical element.
- The tokens available in C++ are:
 - i) Keywords ii) Identifiers
 - iii) Literals iv) Operators
 - v) Punctuators

2. What are keywords? Can keywords be used as identifiers?

- Keywords are the reserved words with which convey specific meaning to C++ compilers.
- They are the essential elements to construct C++ programs. Keywords cannot be used as identifiers.

3. The following constants are of which type?

(i) 39 (ii) 032 (iii) 0XCAFE (iv) 04.14

- (i) 39 - Integer constant (ii) 032 - Octal constant
- (iii) 0XCAFE - Hexadecimal (iv) 04.14 - Floating point constant

4. Write the following real constants into the exponent form:

(i) 23.197 (ii) 7.214 (iii) 0.00005 (iv) 0.319

- (i) $23.197 = 0.23197 \times 10^2 = 0.23197E2$ (ii) $7.214 = 0.7214 \times 10^1 = 0.7214E1$
- (iii) $0.00005 = 0.5 \times 10^{-4} = 0.5E-4$ (iv) $0.319 = 3.19 \times 10^{-1} = 3.19E-1$

5. Assume n=10; what will be result of n++ and --n;?

Operator	Operation	Result
>>	n>>2	0 0 0 0 1 0 1 0 n = 10
		0 0 0 0 1 0 0 0 n >> 2
		0 0 0 0 0 0 1 0
$n \gg 2 = (00000010)_2 = 2_{10}$		

6. Match the following

A	B
(a) Modulus	(1) Tokens
(b) Separators	(2) Remainder of a division
(c) Stream extraction	(3) Punctuators
(d) Lexical Units	(4) get from

A	B
(a) Modulus	(2) Remainder of a division
(b) Separators	(3) Punctuators
(c) Stream extraction	(4) get from
(d) Lexical Units	(1) Tokens

SECTION-C

Short Answers

1. Describe the differences between keywords and identifiers?

Keywords	Identifiers
Keywords are the reserved words which convey specific meaning to C++	Identifiers are the user defined names give to different parts of the C++ program
Keywords cannot be used as an identifier	Identifiers are not reserved
Example: Switch, case, for, if etc...	Example: Name, age, class, etc...

2. Is C++ case sensitive? What is meant by the term “case sensitive”?

- Yes, C++ is case sensitive. The term case sensitive means C++ treats upper and lower case characters differently.

3. Differentiate “=” and “==”.

=	==
It is an assignment operator (=)	It is a relational operator (==)

= is used to assign a value to a variable which is on the left hand side of an assignment statement	== is used to compare two values and the result will be either true or false.
Example a = b (b value is assigned to a)	Example a == b (a value will be compared with b value)

4. What is the use of a header file?

- Header file contains definitions of functions and variables, which is imported or used into any C++ program by using the pre-processor #include statement. Header file have an extension ".h" which contains C++ function declaration and macro definition.

5. Why is main function special?

- The main function is special because it is the entry point for program execution. Example. It plays the role of a door in a house. It is important and compulsory as execution starts from here.

SECTION - D : Explain in detail

1. Write about Binary operators used in C++.

- Binary Operators require two operands.
- Arithmetic operators that perform simple arithmetic operations like addition, subtraction, multiplication, division (+, -, *, %, /) etc. are binary operators which requires minimum of two operands.
- Relational operators are used to determine the relationship between its operands. The relational operators (<, >, >=, <=, ==, !=) are applied on two operands, hence they are binary operators.
- AND, OR (logical operator) both are binary operators. Assignment operator is also a binary operator (+=, -=, *=, /=, %=).

2. What are the types of Errors?

Type of Error	Description
Syntax Error	Syntax errors occur when grammatical rules of C++ are violated.
Semantic Error	Semantic Error occur when there is wrong use of variable / operator / order of execution etc. It is also called as Logical Error.
Run – time error	A run time error occurs during the execution of a program. It is occurs because of some illegal operation that takes place.

CHAPTER - 9 PART-II : DATA TYPES, VARIABLES AND EXPRESSIONS

SECTION-B : Very Short Answers

1. Write a short note on const keyword with an example.

- It is used to declare a constant. It can be used in many constant in C++ program. It can be used with pointers, objects etc...
- Example: const int a=100;

2. What is the use of setw() format manipulator?

- setw() manipulators sets the width of the field assignment for the output. The field width determines the minimum number of characters to be written in output.

3. Why is char often treated as integer data type?

- Character data type is often said to be an integer type, since all the characters are represented in memory by their associated ASCII codes.

4. What is a reference variable? What is its use?

- Reference is an alternative name for a variable. A reference variable is an alias, i.e. another name for an already existing variable. Once a reference is initialized with a variable, either the variable name or the reference name may be used to refer to the variable.

Use:

- Store the address of the variable used at the time of declaration of reference variable.

5. Consider the following C++ statement. Are they equivalent?

`char ch = 67; char ch = 'C';`

- Yes, they are equal

6. What is the difference between 56L and 56?

- 56 is an integer where as 56L is a long integer

7. Determine which of the following are valid constant? And specify their type.

(i) 0.5 (ii) 'Name' (iii) '\t' (iv) 27,822

- (i) 0.5 - Valid constant - Floating point data type
- (ii) 'Name' - Invalid constant
- (iii) '\t' - Valid constant - non graphic character data type
- (iv) 27,822 - Invalid constant

8. Suppose x and y are two double type variable that you want add as integer and assign to an integer variable. Construct a C++ statement to do the above.

```
double x;
double y;
int z = (int) x + (int) y; or int z = (int) (x + y);
```

9. What will be the result of following if num=6 initially.

- (a) cout << num;
- (b) cout << (num==5);

Answer

- a) 6
- b) 0

10. Which of the following two statements are valid? Why? Also write their result.

(i) int a; a = 3,014; (ii) int a; a=(3,014);

- i) It is invalid as comma is not allowed in an integer constant.
 - ii) It is valid. Comma in bracket is allowed.
- The result will be 12.

SECTION-C : Short Answers

1. What are arithmetic operators in C++? Differentiate unary and binary arithmetic operators.

Give example for each of them.

- Arithmetic operators in C++ are used to perform arithmetical calculations in a program.

Unary	Binary
A unary operator requires a single operand	A binary operator requires two operands
Example: Unary +, Unary -, ++, --	Example +, -, *, /, %

2. How relational operators and logical operators are related to one another?

Logical operators uses AND (&&), OR (||) and NOT (!) operators.

These operators yields true or false depending upon the outcome of different relational expressions. Relational expressions are formed by Relational operators.

3. Evaluate the following C++ expressions where x, y, z are integers and m, n are floating point numbers. The value of x = 5, y = 4 and m=2.5;

(i) n = x + y / x; (ii) z = m * x + y; (iii) z *= x * m + x;

Answer

- (i) n = 5.8
- (ii) z = 16
- (iii) z *= x * m + x;
- = 5 x 2.5 + 5
- z * = 17.5
- z = z * 17.5
- = 16 * 17.5
- = 280

CHAPTER - 10 : FLOW OF CONTROL

SECTION-B : Very Short Answers

1. What is a null statement and compound statement?

- The null or empty statement is a statement containing only a semicolon.
- A group of statements means the enclosed by pair of braces { }. This group of statement is called a compound statements or a block.

2. What is selection statement? write it's types?

- The selection statement means the statement(s) executed depends upon a condition. If a condition is true, a true block (a set of statements) is executed otherwise a false block is executed. This statement also called decision statement or selection statement.
- Types are,
 - i) Simple if statement

- ii) if -else statement
- iii) Nested if
- iv) Switch case

3. Correct the following code segment:

```
if (x=1)
    p= 100;
else
    p = 10;
```

Answer:

```
if ( x == 1)
    p = 100;
else
    p = 10;
```

4. What will be the output of the following code:

```
int year;
cin >> year;
if ( year % 400 == 0)
    cout << "Leap";
else
    cout << "Not Leap year";
```

If the input given is (i) 2000 (ii) 2003 (iii) 2010?

Answer

- i) 2000: Leap year
- ii) 2003: Not Leap year
- iii) 2010: Not Leap year

5. What is the output of the following code?

```
for (int i=2; i<=10 ; i+=2)
    cout << i;
```

Answer

2 4 6 8 10

6. Write a for loop that displays the number from 21 to 30.

```
for ( i = 21; i <= 30; i++)
    cout << i;
```

7. Write a while loop that displays numbers 2, 4, 6, 8.....20.

```
int i = 2;
while ( i <= 20 )
{
    cout << i;
    i+=2;
}
```

8. Compare an if and a ? : operator.

If	? :
Multiple statement are used	Only one statement used
Example: if (p == 10) p = 100; else p=10;	Example (p == 10)? p = 100 : p = 10;

SECTION-C : Short Answers

1. Convert the following if-else to a single conditional statement:

```
if (x >= 10)
    a = m + 5;
else
    a = m;
```

Answer

```
a = (x >= 10)? m+5 : m;
```

2. Rewrite the following code so that it is functional:

```
v = 5;
do;
{
```

```
total += v;
cout << total;
while v <= 10
```

Answer

```
int v=5;
do
{
    total += v;
    cout << total;
    v=v+1;
} while ( v <= 10);
```

3. Write a C++ program to print multiplication table of a given number.

```
# include <iostream.h>
int main ( )
{
    int num;
    cout << "Enter Number to find its multiplication table"; cin >> num;
    for (int a = 1; a <= 10; a++)
    {
        cout << num << "*" << a << "=" << num*a << endl;
    }
    return();
}
```

4. Write the syntax and purpose of switch statement.

- The switch statement is a multi – way branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.
- The switch statement replaces multiple if – else sequence.
- The syntax of the switch statement is:

```
switch(expression)
{
    case constant 1:
        statement(s);
        break;
    case constant 2:
        statement(s);
        break;
    .
    .
    .
    default:
        statement(s);
}
```

5. Write a short program to print following series:

(a) 1 4 7 10..... 40

```
# include <iostream.h>
int main ( )
{
    for (int i = 1; i < 40; i = i + 3)
        cout << i << "\t";
}
```

SECTION - D

Explain in detail

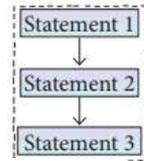
1. Explain control statement with suitable example.

- Control statements are statements that alter the sequence of flow of instructions.

Types of control statements:

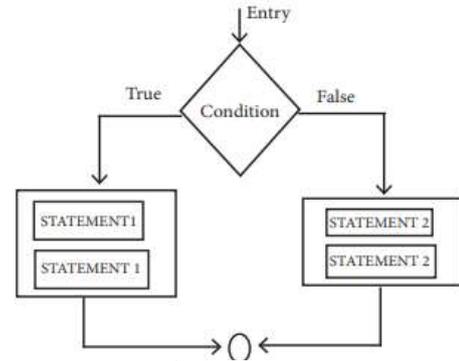
Sequential statement

- The sequential statement are the statements, that are executed one after another only once from top to bottom.
- These statements do not alter the flow of execution. These statements are called as sequential flow statements. They always end with a semicolon (;).



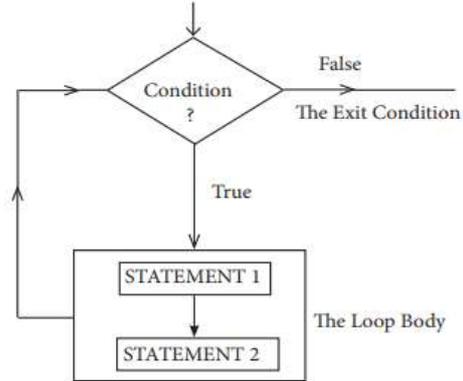
Selection statement

- The selection statement means the statement (s) executed depend upon a condition. If a condition is true, a true block (a set of statements) is executed otherwise a false block is executed.
- This statement is also called decision statement or selection statement because it helps in making decision about which set of statements are to be executed.



Iteration statement

- The iteration statement is a set of statement that are repetitively executed based upon a conditions.
- If a condition evaluates to true, the set of statements (true block) is executed again and again. As soon as the condition becomes false, the repetition stops. This is also known as looping statement or iteration statement.
- The set of statements that are executed again and again is called the body of the loop.
- The condition on which the execution or exit from the loop is called exit-condition or test-condition.



2. What is an entry control loop? Explain any one of the entry controlled loop with suitable example.

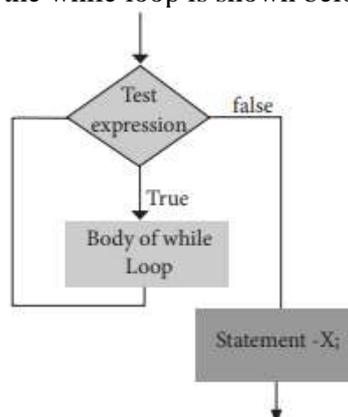
- An entry control loop checks the condition at the time of entry and if condition or expression becomes true then control transfers into the body of the loop, for loop and while loop are the examples of Entry Controlled Loop.
- A while loop is a control flow statement that allows the loop statements to be executed as long as the condition is true.
- The while loop is an entry-controlled loop because the test expression is evaluated before entering into a loop.

- The while loop syntax is:

```

while ( Test expression )
{
    Body of the loop;
}
Statement-x;
  
```

- The control flow and flow chart of the while loop is shown below.



- In while loop, the test expression is evaluated and if the test expression result is true, then the body of the loop is executed and again the control is transferred to the while loop.
- When the test expression result is false the control is transferred to statement-x.

Example Program

```
#include<iostream.h>
using namespace std;
int main ( )
{
    int i=1,sum=0;
    while(i<=10)
    {
        sum=sum+i;
        i++;
    }
    cout<<"The sum of 1 to 10 is " <<sum;
    return 0;
}
Output
The sum of 1 to 10 is 55
```

3. Write a program to find the LCM and GCD of two numbers.

```
#include<iostream>
using namespace std;
int main()
{
    int n1, n2, i, gcd=1, lcm=1;
    cout<<"Enter two numbers you want to find the GCD and LCM of : "<<endl;
    cin>>n1>>n2;
    for( i=1;i<=1000;i++)
    {
        if((n1%i==0) && (n2%i==0))
        {
            gcd=i;
        }
    }
    lcm=(n1*n2)/gcd;
    cout<<"The LCM of the given two number is : "<<lcm<<endl;
    cout<<"The GCD of the given two number is : "<<gcd<<endl;
    return 0;
}
```

Output:

```
Enter two numbers you want to find the GCD and LCM of: 10 20
The LCM of the given two number is 20
The GCD of the given two number is 10
```

4. Write programs to find the sum of the following series:

(a) $x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \frac{x^5}{5!} - \frac{x^6}{6!}$

(b) $x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^n}{n}$

```
a) #include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int x,p,i,j;
    double fact=1.0,ans=0;
    cout<<"Enter the value of x:";
    cin>>x;
```

```

cout<<"\n Enter till what power you want:";
cin>>p;
ans=x;
for(i=2,j=1;i<=p;i++,j++)
{
    fact=fact*i;
    if(i%2==0)
        ans+=(pow(-1,j))*((pow(x,i))/(fact));
}
cout<<"\n The sum of the series is:"<<ans;
return 0;
}

```

Output:

```

Enter the value of x: 3
Enter till what power you want : 4
The sum of the series is :-4.875

```

b)

```

#include <iostream>
#include<math.h>
using namespace std;
int main()
{
    int i,n;
    float x,sum=0;
    cout<<"x+x^2/2+x^3/3+...+x^n/n";
    cout<<"\n Enter value of x:";
    cin>>x;
    cout<<"\n Enter value of n:";
    cin>>n;
    for(i=1;i<=n;++i)
        sum+=pow(x,i)/i;
    cout<<"\n sum="<<sum;
}

```

Output:

```

Enter value of x: 5
Enter value of n: 3
sum = 59.1667

```

5. Write a program to find sum of the series

$$S = 1 + x + x^2 + \dots + x^n$$

```

#include<iostream>
#include<math.h>
int main()
{
    int i,x,n,s=0;
    cout<<"Enter the value if x = ";
    cin>>x;
    cout<<"Enter the Number of terms = ";
    cin>>n;
    for(i=0;i<=n;i++)
        s=s+pow(x,i);
    cout<<"The Sum = " <<s;
    return 0;
}

```

Output:

```

Enter the value if x = 5
Enter the Number of terms = 2
The Sum = 31

```

CHAPTER - 11 : FUNCTIONS

SECTION-B : Very Short Answers

1. Define Functions.

- A large program can typically be split into small sub-program (blocks) called as functions. Each sub-program perform some specific functionality.

2. Write about strlen() function.

- The strlen() takes a null terminated byte string source as its argument and returns its length. The length does not include the null(\0) character.

3. What are importance of void data type.

- To indicate the function does not return a value.
- To declare a generic pointer.

4. What is Parameter and list its types?

- Arguments or parameters are the means to pass values from the calling function to the called function.
- There are two types of parameters:
- **Formal parameters:** The variables used in the function definition as parameters are known as formal parameters.
- **Actual parameters:** The constants, variables or expressions used in the function call are known as actual parameters.

5. Write a note on Local Scope.

- A local variable is defined within a block. A block of code begins and ends with curly braces { }.
- The scope of a local variable is the block in which it is defined.
- A local variable cannot be accessed from outside the block of its declaration.
- A local variable is created upon entry into its block and destroyed upon exit.

SECTION-C : Short Answers

1. What is Built-in functions ?

- C++ provides a rich collection of functions ready to be used for various tasks.
- The tasks to be performed by each of these are already written, debugged and compiled, their definitions alone are grouped and stored in files called header files.
- Such ready – to – use sub programs are called pre – defined functions or built – in functions.

2. What is the difference between isupper() and toupper() functions ?

isupper():

- This function is used to check the given character is uppercase.
- This function will return 1 if true otherwise 0.
- General Form : islower(char c)

toupper():

- This function is used to convert the given character into its uppercase.
- This function will return the upper case equivalent of the given character. If the given character itself is in upper case, the output will be the same.
- General Form: toupper(char c)

3. Write about strcmp() function.

- The strcmp() function takes two arguments: string1 and string2. It compares the contents of string1 and string2 lexicographically.
- The strcmp() function returns a:
 - Positive value if the first differing character in string1 is greater than the corresponding character in string2. (ASCII values are compared)
 - Negative value if the first differing character in string1 is less than the corresponding character in string2.
 - 0 if string1 and string2 are equal.

4. Write short note on pow() function in C++.

- The pow() function returns base raised to the power of exponent. If any argument passed to pow() is long double, the return type is promoted to long double. If not, the return type is double.
- The pow() function takes two arguments:
 - base – the base value
 - exponent – exponent of the base

5. What are the information the prototype provides to the compiler ?

- long fact (int,double)

- The return value of the function is of type long.
- fact is the name of the function.
- The function is called with two arguments:
 - The first argument is of int data type.
 - The second argument is of double data type

6. What is default arguments ? Give example.

- In C++, one can assign default values to the formal parameters of a function prototype. The Default arguments allows to omit some arguments when calling the function.
 - For any missing arguments, compiler uses the values in default arguments for the called function
 - The default value is given in the form of variable initialization.
Example : `void defaultvalue(int n1 = 10, n2 = 100);`
 - The default arguments facilitate the function call statement with partial or no arguments.
Example :

```
defaultvalue (x, y);
defaultvalue (200, 150);
defaultvalue (150);
defaultvalue (x, 150);
```
 - The default values can be included in the function prototype from right to left, i.e., we cannot have a default value for an argument in between the argument list.
Example:

```
void defaultvalue (int n1=10, n2); //invalid prototype.
void defaultvalue (int n1, n2 = 10); //valid prototype.
```

SECTION - D : Explain in detail

1. Explain Call by value method with suitable example.

This method copies the value of an actual parameter into the formal parameter of the function. In this case, changes made to formal parameter within the function will have no effect on the actual parameter.

Example:

```
#include<iostream>
using namespace std;
void display(int x)
{
    int a=x*x;
    cout<<"\n\nThe Value inside display function (a * a):"<<a;
}
int main()
{
    int a;
    cout<<"\n\nExample : Function call by value:";
    cout<<"\n\nEnter the Value for A :";
    cin>>a;
    display(a);
    cout<<"\n\nThe Value inside main function "<<a;
    return(0);
}
```

Output :

```
Example : Function call by value
Enter the Value for A : 5
The Value inside display function (a * a) : 25
The Value inside main function 5
```

2. What is Recursion? Write a program to find the factorial of the given number using recursion.

- A function that calls itself is known as recursive function. And, this technique is known as recursion.
- **Program to find the factorial of the given number using recursion.**

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

```

int main()
{
    int no;
    cout<<"\nEnter a number to find its factorial: ";
    cin >> no;
    cout << "\nFactorial of Number " << no <<" = " << factorial(no);
    return 0;
}
int factorial(int m)
{
    if (m > 1)
    {
        return m*factorial(m-1);
    }
    else
    {
        return 1;
    }
}

```

Output :

```

Enter a number to find its factorial: 5
Factorial of Number 5 = 120

```

3. What are the different forms of function return? Explain with example.

The return statement:

- Returning from the function is done by using the return statement.
- The return statement stops execution and returns to the calling function.
- When a return statement is executed, the function is terminated immediately at that point.
- The return statement is used to return from a function.
- It is categorized as a jump statement because it terminates the execution of the function and transfer the control to the called statement.

Syntax:

```
return expression/variable;
```

Example

```
return(a + b); return(a);
return;
```

The Returning values:

- The functions that return no value is declared as void. The data type of a function is treated as int, if no data type is explicitly mentioned. For example,

For Example:

```
int add (int, int);
add (int, int);
```

- In both prototypes, the return value is int, because by default the return value of a function in C++ is of type int.

Returning Non – integer values:

- A string can also be returned to a calling statement.
- Example

```

#include<iostream>
#include<string.h>
using namespace std;
char *display()
{
    return (“chennai”);
}
int main()
{
    char s[50];

```

```

strcpy(s,display());
cout<<"\nExample:Function with Non Integer Return"<<s;
return(0);
}

```

- **Output :**

Example: Function with Non Integer Return Chennai

4. Explain scope of variable with example.

- Scope refers to the accessibility of a variable.
- There are four types of scopes in C++
 - Local Scope
 - Function Scope
 - File Scope
 - Class Scope

Local Scope:

- A local variable is defined within a block. A block of code begins and ends with curly braces { }.
- The scope of a local variable is the block in which it is defined.
- A local variable cannot be accessed from outside the block of its declaration.
- A local variable is created upon entry into its block and destroyed upon exit;
- Example:

```

int main( )
{
    int a,b; //Local variable
}

```

Function Scope:

- The scope of variable within a function is extended to the function block and all sub-blocks therein.
- The lifetime of a function scope variable is the lifetime of the function block.
- Example:

```

int. sum(intx, int y); //x and y has function scope.

```

File Scope:

- A variable declared above all blocks and functions (including main()) has the scope of a file.
- The lifetime of a file scope variable is the lifetime of a program.
- The file scope variable is also called as global variable.
- Example:

```

#include<iostream>
using namespace std;
int x,y; //x and y are global variable
void main()
{
    .....
}

```

Class Scope:

- Data members declared in a class has the class scope.
- Data members declared in a class can be accessed by all member functions of the class.
- Example:

```

Class example
{
    int x,y; //x and y can be accessed by print() and void():
    void print();
    Void total();
};

```

5. Write a program to accept any integer number and reverse it.

```

#include<iostream>
using namespace std;
int reverse (int n);
int main ()
{
    int n = 0, result = 0;
    cout<< "Enter number:";
    cin >> n;
    result = reverse (n);
    cout<< "Reverse number is:" << result << endl;
}
int reverse (int n)
{
    int temp =0, rev =0;
    while (n! = 0)
    {
        temp = n%10;
        rev = (rev*10) + temp;
        n = n/10;
    }
    return rev;
}

```

Output:

Enter number : 1234
Reverse number: 4321

Chapter - 12 : Arrays and Structures

SECTION-B

Very Short Answers

1. What is Traversal in an Array?

- Accessing each element of an array at least once to perform any operation is known as “Traversal”.

2. What is Strings?

- A string is defined as a sequence of characters where each character may be a letter, number or a symbol. Each element occupies one byte of memory.
- Every string is terminated by a null ('\0', ASCII code 0) character which must be appended at the end of the string.

3. What is the syntax to declare two – dimensional array.

Syntax:

```

data-type array_name[row-size] [col-size];

```

data-type refers to any valid C++ data type
array _ name refers to the name of the 2-D array
row-size refers to the number of rows
col-size refers to the number of columns

4. Define structure .What is its use?

- Structure is a user – defined which has the combination of data items with different data types.
- This allows to group of variables of mixed data types together into a single unit.
- The structure provides a facility to store different data types as a part of the same logical element in one memory chunk adjacent to each other.

5. What is the error in the following structure definition.

```

struct employee{

```

```

    int eno;
    char ename[20];
    char dept;
}Employee e1,e2;

```

Answer:

- Structure tag should be start with uppercase
- Structure not terminated with semicolon
- Space between datatype and variable

```

struct Employee
{
    int eno;
    char ename[20];
    char dept;
};
Employee e1,e2;

```

SECTION-C

Short Answers

1. Define an Array ? What are the types?

- “An array is a collection of variables of the same type that are referenced by a common name”.
- An array is also a derived datatype in C++.
- There are different types of arrays used in C++. They are:
 - One – dimensional arrays
 - Two – dimensional arrays
 - Multi – dimensional arrays

2. Write note an Array of strings.

- An array of strings is a two – dimensional character array.
- The size of the first Index (rows) denotes the number of strings and the size of the second index (columns) denotes the maximum length of each string.
- Usually, array of strings are declared in such a way to accommodate the null character at the end of each string.
- For example, the 2 – D array has the declaration:

```
char name [7][10];
```
- In the above declaration,

```
No. of rows = 7;
No. of columns =10;
```

We can store 7 strings each of maximum length 10 characters.

3. The following code sums up the total of all students name starting with ‘S’ and display it.

Fill in the blanks with required statements.

```

struct student {int exam no,lang,eng,phy,che,mat,csc,total;char name[15];};
int main()
{
    student s[20];
    for(int i=0;i<20;i++)
    { ..... //accept student details }
    for(int i=0;i<20;i++)
    {
        ..... //check for name starts with letter “S”
        ..... // display the detail of the checked name
    }
    return 0;
}

```

Answer

//accept student details

```

    cout<<"Enter Exam number"<<endl;
    cin>>s[i].examno;

```

```

cout<<"Enter Student name<<endl;
cin>>s[i].name;
cout<<"Enter language mark"<<endl;
cin>>s[i].lang;
cout<<"Enter English mark "<<endl;
cin>>s[i].eng;
cout<<"Enter Physics mark "<<endl;
cin>>s[i].phy;
cout<<"Enter Chemistry mark "<<endl;
cin>>s[i].che;
cout<<"Enter Maths mark "<<endl;
cin>>s[i].mat;
cout<<"Enter Computer Science mark "<<endl;
cin>>s[i].csc;
//check for name starts with letter "S"
if(s[i].name[0]=='S')
// display the detail of the checked name
{
    cout<<"Exam Number:"<<s[i].examno<<endl;
    cout<<"Name:"<<s[i].name<<endl;
    cout<<"Language Mark:"<<s[i].lang<<endl;
    cout<<"English Mark:"<<s[i].eng<<endl;
    cout<<"Physics Mark:"<<s[i].phy<<endl;
    cout<<"Chemistry Mark:"<<s[i].che<<endl;
    cout<<"Maths Mark:"<<s[i].mat<<endl;
    cout<<"Computer Science Mark:"<<s[i].csc<<endl;
    total=s[i].lang+s[i].eng+s[i].phy+s[i].che+s[i].mat+s[i].csc;
    cout<<"Total:"<<total<<endl;
}

```

4. How to access members of a structure? Give example.

- Once objects of a structure type are declared, their members can be accessed directly.
- The syntax for that is using a dot(.) between the object name and the member name.
Example: s1.name;
- If the members are a pointer type, then is used to access the members.
- Let name be a character pointer in student like char *name
- It can be accessed s1 → name,

```

struct Student
{
    long rollno;
    int age;
    float weight;
};
Student s1;

```

5. What is called anonymous structure .Give an example

- A structure without a name/tag is called anonymous structure.

```

Example:
struct
{
    long rollno;
    int age;
    float weight;
}Student;

```

SECTION - D

Explain in detail

1. Write a C++ program to find the difference between two matrix.

```
#include<iostream.h>
```

```

void main( )
{
    int m1[2][2], m2[2][2],s[2][2];
    int i,j;
    cout<<"Enter the First Matrix Values:";
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            cin>>m1[i][j];
        }
    }
    cout<<"Enter the Second Matrix Values:";
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            cin>>m2[i][j];
        }
    }
    cout<<"Difference Between Two Matrix:";
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            s[i][j]=m1[i][j]+m2[i][j];
            cout<<s[i][j];
        }
    }
    cout<<endl;
}

```

Output:

Enter the First Matrix Values:

8
3
5
2

Enter the First Matrix Values:

5
1
2
1

Difference Between Two Matrices:

3 2
3 1

2. Write a C++ program to add two distances using the following structure definition

```

struct Distance{
    int feet;
    float inch;
    }d1 , d2, sum;

```

```

#include<iostream.h>

```

```

struct Distance

```

```

{
    int feet;
    float inch;
    }d1,d2,sum;
void main( )

```

```

{
    cout<<"Enter First Distance:"<<endl;
    cout<<"Enter Feet:";
    cin>>d1.feet;
    cout<<"Enter Inch:";
    cin>>d1.inch;
    cout<<"Enter Second Distance:"<<endl;
    cout<<"Enter Feet:";
    cin>>d2.feet;
    cout<<"Enter Inch:";
    cin>>d2.inch;
    sum.feet=d1.feet+d2.feet;
    sum.inch=d1.inch+d2.inch;
    if(sum.inch>12)
    {
        ++sum.feet;
        sum.inch=12;
    }
    cout<<"Sum of the Distance:"<<sum.feet<<" feet \t"<<sum.inch<<" inches"<<endl;
}

```

Output:

```

Enter First Distance:
Enter feet: 6
Enter inch: 3.4
Enter Second Distance:
Enter feet: 5
Enter inch: 10.2
Sum of the Distance = 12 feet 1.6 inches

```

3. Write the output of the following c++ program

```

#include<iostream>           int main()
#include<stdio>              { cout<< "Details of Book No " << 1 << "\n";
#include<string>             cout<< "-----\n";
#include<conio>              cout<< "Book Name : "<<strcpy(a[0].name,"Programming ")<<endl;
using namespace std;        cout<< "Book Author : "<<strcpy(a[0].author,"Dromy")<<endl;
struct books {              cout<< "\nDetails of Book No " << 2 << "\n";
char name[20], author[20];  cout<< "-----\n";
} a[2];                     cout<< "Book Name : "<<strcpy(a[1].name,"C++programming" )<<endl;
                             cout<< "Book Author : "<<strcpy(a[1].author,"BjarneStroustrup ")<<endl;
                             cout<< "\n\n";
                             cout<< "===== \n";
                             cout<< " S.No\t| Book Name\t|author\n";
                             cout<< "-----";
                             for (int i = 0; i < 2; i++) {
                             cout<< "\n " << i + 1 << "\t| " << a[i].name << "\t| " << a[i].author;
                             }
                             cout<< "\n===== ";
                             return 0;
                             }

```

Output:

```

Details of Book No. 1
-----
Book name : Programming
Book author : Dromy

Details of Book No. 2
-----
Book Name : C++ Programming
Book Author : Bjarne Stroustrup

```

```

=====
S.No | Book Name | Author
=====
1 | Programming | Dromy
2 | C++ Programming | Bjarne Stroustrup
=====

```

4. Write the output of the following c++ program

```

#include <iostream>
#include <string>

```

```

using namespace std;
struct student
{
    int roll_no;
    char name[10];
    long phone_number;
};
int main(){
    student p1 = {1,"Brown",123443},p2;
    p2.roll_no = 2;
    strcpy(p2.name ,"Sam");
    p2.phone_number = 1234567822;
    cout<< "First Student" <<endl;
    cout<< "roll no : " << p1.roll_no <<endl<< "name : " << p1.name <<endl;
    cout<< "phone no : " << p1.phone_number <<endl;
    cout<< "Second Student" <<endl;
    cout<< "roll no : " << p2.roll_no <<endl<< "name : " << p2.name <<endl;
    cout<< "phone no : " << p2.phone_number <<endl;
    return 0;
}

```

Output:

```

First Student
roll no: 1
name: Brown
phone no: 123443
Second Student
roll no: 2
name: Sam
phone no: 1234567822

```

5. Debug the error in the following program

```

#include <iostream.h>
struct PersonRec
{
    char lastName[10];
    char firstName[10];
    int age;
}
PersonRec peopleArray[10];
void main()
{
    PersonRecord people;
    for (i = 0; i < 10; i++)
    {
        cout<<people.firstName<< ' '
        <<people.lastName <<people.age;
    }
    for (int i = 0; i < 10; i++)
    {
        cout<< "Enter first name: ";
        cin<<peop[i].firstName;
        cout<< "Enter last name: ";
        cin>>peop[i].lastName;
        cout<< "Enter age: ";
        cin>> people[i].age;}
}

```

After Debug the error:

```

#include<iostream.h>
struct PersonRec
{
    char lastName[10];
    char firstName[10];
    int age;
};
PersonRec people[10];
void loadarray( );
void main()
{
    int i;
    loadarray( );
    for(i=0;i<10;i++)
    {
        cout<<people[i].firstName;
        cout<<people[i].lastName;
        cout<<people[i].age;
    }
}
void loadarray( )
{
    for(i=0;i<10;i++)
    {
        cout<<"Enter First Name";
        cin>>people[i].firstName;
        cout<<"Enter Last Name";
        cin>>people[i].lastName;
        cout<<"Enter Age";
        cin>>people[i].age;
    }
}

```

Chapter-13 Introduction to Object Oriented Programming Techniques

SECTION-B : Very Short Answers

1. How is modular programming different from procedural programming paradigm?

Modular programming:

- Emphasis on algorithm rather than data.
- Programs are divided into individual modules.
- Each modules are independent of each other and have their own local data.
- Modules can work with its own data as well as with the data passed to it.

Procedural programming:

- Programs are organized in the form of subroutines or sub programs.
- All data items are global.
- Suitable for small sized software application.
- Difficult to maintain and enhance the program code as any change in data type needs to be propagated to all subroutines that use the same data type.

2. Differentiate classes and objects.

Class:

- Class is a blue print or template from which objects are created.
- Class doesn't allocate memory when it is created.
- Class is a logical entity.

Object:

- Object is an instance of a class.
- Objects allocate memory when it is created.
- Object is a physical entity.

3. What is polymorphism?

- Polymorphism is the ability of a message or function to be displayed in more than one form.

4. How is encapsulation and abstraction are interrelated?

- Abstraction means giving only essential things and hiding unnecessary details.
- Encapsulation is the binding of data members and methods together in a capsule to avoid accidental changes to data from external users, i.e., encapsulation is the bundling of related algorithms and data.

5. Write the disadvantages of OOP.

Size: Object Oriented Programs are much larger than other programs.

Effort: Object Oriented Programs require a lot of work to create.

Speed: Object Oriented Programs are slower than other programs, because of their size.

SECTION-C : Short Answers

1. What is paradigm ?Mention the different types of paradigm.

- Paradigm means organizing principle of a program. It is an approach to programming.
- There are different approaches available for problem solving using computer.
- They are Procedural programming, Modular Programming and Object Oriented Programming.

2. Write a note on the features of procedural programming.

Important features of procedural programming

- Programs are organized in the form of subroutines or sub programs
- All data items are global
- Suitable for small sized software application
- Difficult to maintain and enhance the program code as any change in data type needs to be propagated to all subroutines that use the same data type. This is time consuming.
- Example: FORTRAN and COBOL.

3. List some of the features of modular programming

Important features of Modular programming:

- Emphasis on algorithm rather than data
- Programs are divided into individual modules
- Each modules are independent of each other and have their own local data
- Modules can work with its own data as well as with the data passed to it.
- Example: Pascal and C.

4. What do you mean by modularization and software reuse?

- Modularization : where the program can be decomposed into modules.
- Software reuse : where a program can be composed from existing and new modules.

5. Define information hiding.

- Encapsulation is the most striking feature of a class.
- The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it.
- This encapsulation of data from direct access by the program is called data hiding or information hiding.

SECTION - D : Explain in detail

1. Write the differences between Object Oriented Programming and procedural programming.

Object Oriented Programming:

- Emphasizes on data rather than algorithm.
- Data abstraction is introduced in addition to procedural abstraction.
- Data and its associated operations are grouped in to single unit.
- Programs are designed around the data being operated.
- Example: C++, Java, VB.Net, Python

Procedural Programming:

- Programs are organized in the form of subroutines or sub programs.
- All data items are global.
- Suitable for small sized software application.
- Difficult to maintain and enhance the program code as any change in data type needs to be propagated to all subroutines that use the same data type.
- Example: FORTRAN and COBOL

2. What are the advantages of OOPs?

- Re-usability :
 - “Write once and use it multiple times” you can achieve this by using class.
- Redundancy:
 - Inheritance is the good feature for data redundancy.
 - If you need a same functionality in multiple class you can write a common class for the same functionality and inherit that class to sub class.
- Easy Maintenance :
 - It is easy to maintain and modify existing code as new objects can be created with small differences to existing ones.
- Security :
 - Using data hiding and abstraction only necessary data will be provided thus maintains the security of data.

3. Write a note on the basic concepts that supports OOPs?

Basic concepts that supports OOPs

- Modularization:
 - The program can be decomposed into modules.
- Software re-use:
 - A program can be composed from existing and new modules.
- Encapsulation:
 - The mechanism by which the data and functions are bound together into a single unit is known as Encapsulation. It implements abstraction.
 - Encapsulation is about binding the data variables and functions together in class. It can also be called data binding.
 - The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it.
 - This encapsulation of data from direct access by the program is called data hiding or information hiding.
- Data Abstraction:
 - Abstraction refers to showing only the essential features without revealing background details.
 - Classes use the concept of abstraction to define a list of abstract attributes and function which operate on these attributes.
 - The functions that operate on these data are called methods or member function.
- Modularity:

- Modularity is designing a system that is divided into a set of functional units (named modules) that can be composed into a larger application.
- Inheritance:
 - Inheritance is the technique of building new classes (derived class) from an existing Class (base class). The most important advantage of inheritance is code reusability.
- Polymorphism:
 - Polymorphism is the ability of a message or function to be displayed in more than one form.

Chapter-14 : Classes and objects

SECTION-B : Very Short Answers

1. What are called members?

- Class comprises of members
- Members are classified as Data members and Member functions.

2. Differentiate structure and class though both are user defined data type.

Structure	Class
Members of structure are public by default	Members of class is private by default
Structure does not support inheritance	Class can create a sub class by inheritance

3. What is the difference between the class and object in terms of oop?

Class	Object
It is a data type	It is an instance of class
It generates objects	It gives life to class
It does not occupy memory location	It occupies memory location

4. Why it is considered as a good practice to define a constructor though compiler can automatically generate a constructor ?

- To allocate memory space to the object
- To initialize the data member of the class object

5. Write down the importance of destructor.

- To free the resources that the object may have acquired during its lifetime.
- Destruction function removes the memory of an object which was allocated by the constructor at the time of creating object.

SECTION-C : Short Answers

1. Rewrite the following program after removing the syntax errors if any and underline the errors:

```
#include<iostream>
$include<stdio>
class mystud
{
    int studid =1001;
    char name[20];
    public
    mystud( ) { }
    void register ( )
    { cin>>stdid; gets(name);
    }
    void display ( )
    { cout<<studid<<": "<<name<<endl;
    }
}
int main()
{
    mystud MS;
    register.MS();
    MS.display();
}
```

Answer:

```
#include<iostream>
#include<stdio.h>
using namespace std;
class mystud
{
    int studid =1001;
    char name[20];
    public:
    mystud( )
    {
    }
    void register1( )
    {
        cin>>stdid;
        gets(name);
    }
    void display ( )
    {
        cout<<studid<<": "<<name<<endl;
    }
}
int main()
{
    mystud MS;
    MS.register1();
    MS.display( );
}
```

2. Write with example how will you dynamically initialize objects?

- When the initial values are provided during runtime then it is called dynamic initialization.

```
#include<iostream>
using namespace std;
class X
{
    int n;
public:
    X(int p)
    {
        n=p;
    }
    void disp()
    {
        cout<<"\n Roll number:- " <<n;
    }
};
int main()
{
    int a ;
    cout<<"\nEnter the Roll Number";
    cin>>a;
    X x(a);
    x.disp();
    return 0;
}
```

Output:

```
Enter the Roll Number 1201
Roll number:- 1201
```

3. What are advantages of declaring constructors and destructor under public accessibility?

- Advantages of declaring constructors and destructor under public accessibility is that its object can be created in any function.

4. Given the following C++ code, answer the questions (i) & (ii).

```
class TestMeOut
{
public:
    ~TestMeOut() //Function 1
    {
        cout<<"Leaving the exam hall"<<endl;
    }
    TestMeOut() //Function 2
    {
        cout<<"Appearing for exam"<<endl;
    }
    void MyWork() //Function 3
    {
        cout<<"Answering"<<endl;
    }
};
```

(i) In Object Oriented Programming, what is Function 1 referred as and when does it get invoked / called ?

(ii) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked / called ?

Answer:

- Destructor, when an instance of a class goes out of scope the destructor gets invoked/called.
- Constructor, when an instance of a class comes into scope in constructor gets invoked/called.

SECTION - D : Explain in detail

1. Mention the differences between constructor and destructor

Constructor	Destructor
It allocates that memory to an object	It deallocates the memory of an object
It can be declared as class-name(arguments) { };	It can be declared as ~class-name(no arguments) { };
It accepts arguments	It does not accepts any argument.
Constructor can be overloaded	It cannot be overloaded
There can be multiple constructor in the class	Only one destructor in the class

2. Define a class RESORT with the following description in C++ :

Private members:

Rno // Data member to store room number

Name //Data member to store user name

Charges //Data member to store per day charge

Days //Data member to store the number of days

Compute()/*A function to calculate total amount as Days * Charges and if the total amount exceeds 11000 then total amount is 1.02 * Days *Charges */

Public member:

GetInfo() /* Function to Read the information like name , room no, charges and days*/

DispInfo()/* Function to display all entered details and total amount calculated using COMPUTE function*/

Answer:

```
#include<iostream>
#include<stdio.h>
using namespace std;
class RESORT
{
private:
    int Rno;
    char name [20];
    float charges;
    int days;
    float compute();
public:
    void getinfo();
    void dispinfo();
};
void RESORT :: getinfo()
{
    cout << "Enter Registration number: "; cin >> Rno.;
    cout << "\n Enter name: "; gets(name);
    cout << "\n Enter per day charges: "; cin >> charges;
    cout << "\n Enter number of days: "; cin >> days;
}
void RESORT :: dispinfo()
{
    cout << "\n1. Registration number:" << Rno << "\n2. Name: "; puts(name);
    cout << "\n3. charges per day:" << charges << "\n4. Number of days:" << days;
    cout << "\n5. Amount:" << compute();
}
float RESORT :: compute()
{
    float amount=0.0;
    amount = charges*days;
    if (amount> 11000)
```

```

        amount = 1.02*days*charges;
    return amount;
}
void main()
{
    RESORT obj;
    obj.getinfo();
    obj.dispinfo();
}

```

3. Write the output of the following

```

#include<iostream>
using namespace std;
class student
{
    int rno, marks;
public:
    student(int r,int m)
    {
        cout<<"Constructor "<<endl;
        rno=r;
        marks=m;
    }
    void printdet()
    {
        marks=marks+30;
        cout<<"Name: Bharathi"<<endl;
        cout<<"Roll no : "<<rno<<"\n";
        cout<<"Marks : "<<marks<<endl;
    }
};
int main()
{
    student s(14,70);
    s.printdet();
    cout<< "Back to Main";
    return 0;
}

```

Output:

```

Constructor
Name: Bharathi
Roll no:14
Marks:100
Back to Main

```

Chapter-15 : Polymorphism

SECTION-B : Very Short Answers

1. What is function overloading?

- Two or more functions in the same scope that share the same name but their parameters are different. The functions that share the same name are said to be overloaded and the process is called function overloading.

2. List the operators that cannot be overloaded.

- Operator that are not overloaded are as follows,
 - scope operator ::
 - size of
 - member selector
 - member pointer selector
 - ternary operator ?:

3. class add{int x; public: add(int)}; Write an outline definition for the constructor.

```

add :: add(int y)
{
    y=x;
}

```

4. Does the return type of a function help in overloading a function?

- No, the return type of a function not help in overloading a function.

5. What is the use of overloading a function?

- Function overloading is not only implementing polymorphism but also reduces the number of comparisons in a program and makes the program to execute faster.
- It also helps the programmer by reducing the number of function names to be remembered.

SECTION-C : Short Answers

1. What are the rules for function overloading?

- The overloaded function must differ in the number of its arguments or data types
- The return type of overloaded functions are not considered for overloading same data type
- The default arguments of overloaded functions are not considered as part of the parameter list in function overloading.

2. How does a compiler decide as to which function should be invoked when there are many functions? Give an example.

- When the overloaded function called, the compiler determines the most appropriate definition to use, by comparing the argument types you have used to call the function with the parameter types specified in the definitions.
- The process of selecting the most appropriate overloaded function or operator is called overload resolution.
- Example:
 - float area_circle(float radius)
 - float area_triangle(float half,floatbase,float height)
 - float area_rectangle(float length , float breadth)

3. What is operator overloading? Give some examples of operators which can be overloaded.

- The mechanism of giving special meaning to an operator is known as operator overloading.
- Operators that are to be overloaded are as +, ++, -, - -, +=, -=, *, <, >.

4. Discuss the benefits of constructor overloading ?

- Constructor overloading provides flexibility of creating multiple type of objects for a class.
- Compiler identifies a given member function is a constructor by its name and the return type.
- There are multiple constructors present, argument to the constructor should also be passed while creating an object.

5. class sale (int cost, discount ;public: sale(sale &);

Write a non inline definition for constructor specified;

```
sale :: (sale &x)
{
    cost=x.cost;
    discount=x.discount;
}
```

SECTION - D : Explain in detail

1. What are the rules for operator overloading?

- Precedence and Associativity of an operator cannot be changed.
- No new operators can be created, only existing operators can be overloaded.
- Cannot redefine the meaning of an operator's procedure. You cannot change how integers are added. Only additional functions can be given to an operator
- Overloaded operators cannot have default arguments.
- When binary operators are overloaded, the left hand object must be an object of the relevant class

2. Answer the question (i) to (v) after going through the following class.

```
class Book
{
    int BookCode ;
    char Bookname[20];
    float fees;
public:
    Book() //Function 1
    {
        fees=1000;
        BookCode=1;
        strcpy(Bookname,"C++");
    }
}
```

```

void display(float C) //Function 2
{
    cout<<BookCode<<":"<<Bookname<<":"<<fees<<endl;
}
~Book() //Function 3
{
    cout<<"End of Book Object"<<endl;
}
Book (int SC,char S[ ],float F) ; //Function 4
};

```

- (i) In the above program, what are Function 1 and Function 4 combined together referred as?
(ii) Which concept is illustrated by Function3? When is this function called/ invoked?
(iii) What is the use of Function3?
(iv) Write the statements in main to invoke function1 and function2
(v) Write the definition for Function4 .

Answer:

- i) Constructor overloading
ii) Destructor, when instance of a class goes out of scope, the function gets called.
iii) It deallocates the memory allocated by the constructor Book()
iv) int main()

```

{
    Book B1;
    B1.display(10,5);
}

```

- v) Book (int SC,char S[],float F)

```

{
    BookCode=SC;
    strcpy(Bookname,S);
    fees=F;
}

```

3. Write the output of the following program

```

include<iostream>
using namespace std;
class Seminar
{ int Time;
public:
    Seminar()
    { Time=30;cout<<"Seminar starts now"<<endl;
    }
void Lecture()
{ cout<<"Lectures in the seminar on"<<endl;
}
Seminar(int Duration)
{ Time=Duration;cout<<"Welcome to Seminar "<<endl;
}
Seminar(Seminar &D)
{ Time=D.Time;cout<<"Recap of Previous Seminar Content "<<endl;
}
~Seminar()
{ cout<<"Vote of thanks"<<endl;
}
};
int main()
{
    Seminar s1,s2(2),s3(s2);
    s1.Lecture();
    return 0;
}

```

Answer:

```

Seminar starts now
Welcome to Seminar
Recap of Previous Seminar Content
Lectures in the seminar on
Vote of thanks
Vote of thanks
Vote of thanks

```

4. Answer the questions based on the following program

```
#include<iostream>
#include<string.h>
using namespace std;
class comp {
public:
char s[10];
void getstring(char str[10])
{ strcpy(s,str); }
void operator==(comp);
};
void comp::operator==(comp ob)
{ if(strcmp(s,ob.s)==0)
cout<<"\nStrings are Equal";
else
cout<<"\nStrings are not Equal"; }
int main()
{ comp ob, ob1;
char string1[10], string2[10];
cout<<"Enter First String:";
cin>>string1;
ob.getstring(string1);
cout<<"\nEnter Second String:";
cin>>string2;
ob1.getstring(string2);
ob==ob1;
return 0; }
```

- (i) Mention the objects which will have the scope till the end of the program.
 - (ii) Name the object which gets destroyed in between the program
 - (iii) Name the operator which is over loaded and write the statement that invokes it.
 - (iv) Write out the prototype of the overloaded member function
 - (v) What types of operands are used for the overloaded operator?
 - (vi) Which constructor will get executed in the above program?
- Write the output of the program

Answer:

- i) ob,ob1
- ii) ob
- iii) ob == ob1
- iv) void operator == (comp);
- v) string
- vi) default constructor will get executed
Enter first string: COMPUTER
Enter second string: PROGRAM
Strings are not equal

Chapter-16 : Inheritance

SECTION-B : Very Short Answers

1. What is inheritance?

- The mechanism of deriving new class from an existing class is known as inheritance. Existing class is called Base class and the derived class is called Sub class.

2. What is a base class?

- The existing class is called as base class.
- A class used for creating a new class is called a super class or base class.

3. Why derived class is called power packed class?

- It can add additional attributes and methods and thus enhance its functionality.

4. In what multilevel and multiple inheritance differ though both contains many base class?

- Multiple inheritance is an inheritance type where a class inherits from more than one base class.
- Multilevel inheritance is an inheritance that inherits from a derived class, making that derived class as a base class for a new class.
- Multiple inheritance is not widely used. Multilevel inheritance is widely used.

5. What is the difference between public and private visibility mode?

Public visibility mode:

- When a base class is inherited with public visibility mode , the protected members of the base class will be inherited as protected members of the derived class and the public members of the base class will be inherited as public members of the derived class.

Private visibility mode:

- When a base class is inherited with private visibility mode the public and protected members of the base class become 'private' members of the derived class.

SECTION-C : Short Answer

1. What are the points to be noted while deriving a new class?

- i) The keyword class has to be used
- ii) The name of the derived class is to be given after the keyword class
- iii) A single colon (:)

- iv) The type of derivation (the visibility mode), namely private, public or protected. If no visibility mode is specified ,then by default the visibility mode is considered as private.
- v) The name of the base class(parent class), if more than one base class, then it can be given separated by comma.

2. What is difference between the members present in the private visibility mode and the members present in the public visibility mode

Private visibility mode:

- The private member of a class cannot be inherited.
- In privately derived class, the public and the protected members of the base class become private in derived class

Public visibility mode:

- The public member of a class can be accessed anywhere in the program.
- In publicly derived class, the public members of the base class remain public and protected members of base class remain protected in derived class.

3. What is the difference between polymorphism and inheritance though are used for reusability of code?

- i) Polymorphism reuse methods by applying them in similar kind of objects.
- ii) Inheritance reuse and extends class definitions, creating new class from an existing class.
- iii) Inheritance permits subclasses to reuse the fields and methods of super classes.

4. What do you mean by overriding?

- When a derived class member function has the same name as that of its base class member function, the derived class member function shadows/hides the base class's inherited function. This situation is called function overriding.
- This can be resolved by giving the base class name followed by :: and the member function name.

5. Write some facts about the execution of constructors and destructors in inheritance

- Constructors and destructors of the base class are not inherited but during the creation of an object for derived class the constructors of base class will automatically be invoked.
- The constructors are executed in the order of inherited class i.e., from base constructor to derived. The destructors are executed in the reverse order.

SECTION - D : Explain in detail

1. Explain the different types of inheritance

Different types of inheritance

Single Inheritance

- When a derived class inherits only from one base class, it is known as single inheritance

Multiple Inheritance

- When a derived class inherits from multiple base classes it is known as multiple inheritance

Hierarchical inheritance

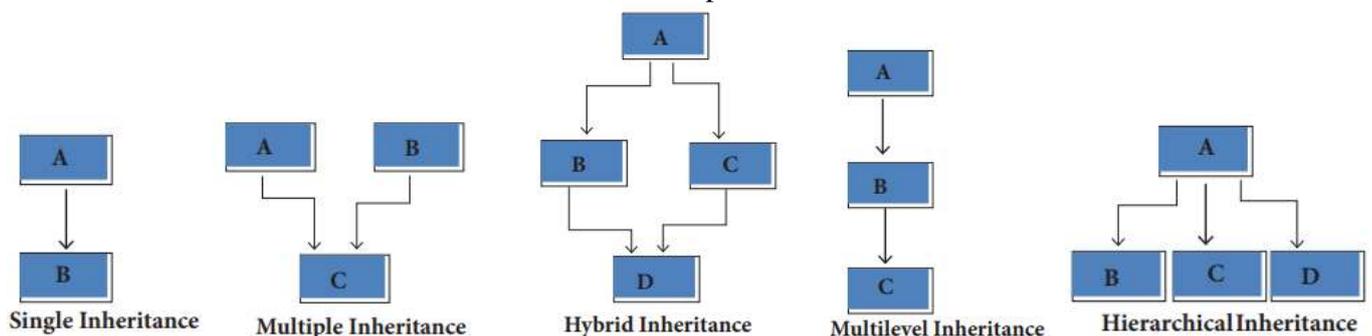
- When more than one derived classes are created from a single base class , it is known as Hierarchical inheritance.

Multilevel Inheritance

- The transitive nature of inheritance is reflected by this form of inheritance. When a class is derived from a class which is a derived class – then it is referred to as multilevel inheritance.

Hybrid inheritance

- When there is a combination of more than one type of inheritance, it is known as hybrid inheritance. Hence, it may be a combination of Multilevel and Multiple inheritance or Hierarchical and Multilevel inheritance or Hierarchical, Multilevel and Multiple inheritance



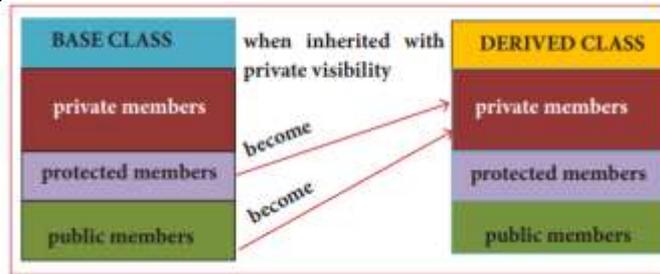
2. Explain the different visibility mode through pictorial representation

Visibility mode

- The accessibility of base class by the derived class is controlled by visibility modes. The three visibility modes are private, protected and public. The default visibility mode is private.

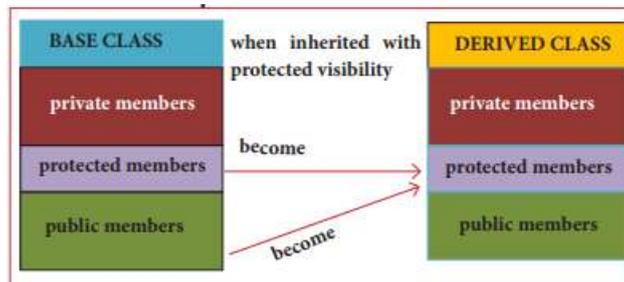
Private visibility mode

- When a base class is inherited with private visibility mode the public and protected members of the base class become 'private' members of the derived class



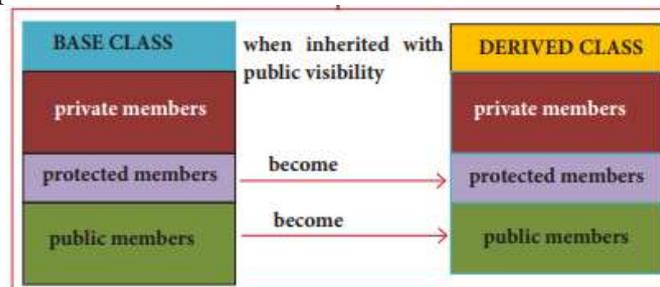
Protected visibility mode

- When a base class is inherited with protected visibility mode the protected and public members of the base class become 'protected' members of the derived class



Public visibility mode

- When a base class is inherited with public visibility mode, the protected members of the base class will be inherited as protected members of the derived class and the public members of the base class will be inherited as public members of the derived class.



3. Consider the following c++ code and answer the questions

```
class Personal
{
    int Class,Rno;
    char Section;
protected:
    char Name[20];
public:
    personal();
    void pentry();
    void Pdisplay();
};
class Marks:private Personal
{
    float M{5};
protected:
    char Grade[5];
public:
    Marks();
```

```

    void Mentry();
    void Mdisplay();
};
class Result:public Marks
{
    float Total,Agg;
public:
    char FinalGrade, Commence[20];
    Result();
    void Rcalculate();
    void Rdisplay();
};

```

- 3.1. Which type of Inheritance is shown in the program?
- 3.2. Specify the visibility mode of base classes.
- 3.3. Give the sequence of Constructor / Destructor Invocation when object of class Result is created.
- 3.4. Name the base class(/es) and derived class (/es).
- 3.5 Give number of bytes to be occupied by the object of the following class:
(a) Personal (b) Marks (c) Result
- 3.6. Write the names of data members accessible from the object of class Result.
- 3.7. Write the names of all member functions accessible from the object of class Result.
- 3.8. Write the names of all members accessible from member functions of class Result.

Answer

- 3.1. Multilevel Inheritance
- 3.2. Public
- 3.3. Constructor Invocation
 - Constructor of base class Personal
 - Constructor of derived class Marks
 - Constructor of derived class Result
Destructor Invocation
 - Destructor of derived class Result
 - Destructor of derived class Marks
 - Destructor of base class Personal
- 3.4. Base Classes
Personal
Derived Classes
Marks
Result
- 3.5. Personal - 29 Bytes
Marks - 54 Bytes
Result - 83 Bytes
- 3.6. FinalGrade, Commence
- 3.7. Mentry()
Mdisplay()
Rcalculate()
Rdisplay()
- 3.8. Total
Agg
Grade
FinalGrade
Commence

4. Write the output of the following program

```

#include<iostream>
using namespace std;
class A
{
    protected:

```

```

    int x;
public:
    void show()
    {
        cout<<"x = "<<x<<endl;
    }
    A()
    {
        cout<<endl<<" I am class A "<<endl;
    }
    ~A()
    {
        cout<<endl<<" Bye ";
    }
};

```

```

class B : public A
{
    protected:
        int y;
    public:
        B(int x1, int y1)
        {
            x = x1;
            y = y1;
        }
        B()
        {
            cout<<endl<<" I am class
B "<<endl;
        }
        ~B()
        {
            cout<<endl<<" Bye ";
        }
        void show()

```

```

int main()
{
    A objA;
    B objB(30, 20);
    objB.show();
    return 0;
}

```

Answer:

```

I am class A
I am class B
x = 30
y = 20
Bye
Bye

```

5. Debug the following program

```

#include(iostream.h)
#include<conio.h>
class A()
{
public;
    int a1,a2:a3;
    void getdata[]
    { a1=15; a2=13; a3=13; } }
class B:: public A()
{
PUBLIC
    voidfunc()
    {
        int b1:b2:b3;
        A::getdata[];
        b1=a1;
        b2=a2;
        a3=a3;
        cout<<b1<<'t'<<b2<<'t'<<b3; }
}
void main()
{
    B der;
    -der1:func();
}

```

Answer:

```

#include<iostream.h>
#include<conio.h>
using namespace std;

```

```

class A
{
    public:
    int a1,a2,a3;
    void getdata()
    {
        a1=15;
        a2=14;
        a3=13;
    }
};
class B: public A
{
    public:
    void func()
    {
        int b1,b2,b3;
        getdata();
        b1=a1;
        b2=a2;
        b3=a3;
        cout<<b1<<"\t"<<b2<<"\t"<<b3;
    }
};
int main()
{
    clrscr();
    B der;
    der.func();
    getch();
}

```

Chapter-17 Computer Ethics And Cyber Security

SECTION-B : Very Short Answers

1. What is harvesting?

- A person or program collects login and password information from a legitimate user to illegally gain access to others accounts.

2. What are Warez?

- Commercial programs that are made and available to the public illegally are often called Warez.

3. Write a short note on cracking.

- Cracking means trying to get into computer system in order to steal, corrupt, or illegitimately view data.

4. Write two types of cyber attacks.

- Virus
- Worms

5. What is a Cookie?

- It is a small piece of data sent from a website and stored on the user's computer memory by the user's web browser while the user is browsing Internet.

SECTION-C : Short Answers

1. What is the role of firewalls?

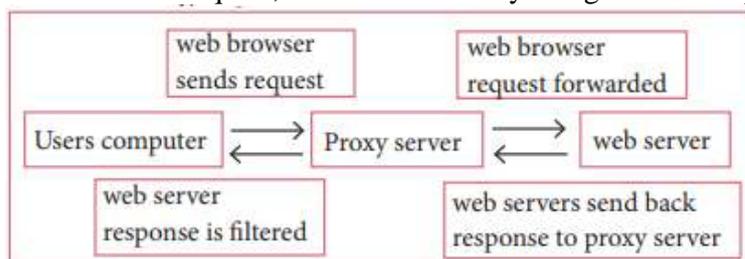
- A firewall is a computer network security based system that monitors and controls incoming and outgoing network traffic based on predefined security rules.
- A firewall commonly establishes a block between a trusted internal computer network and entrusted computer outside the network.

2. Write about encryption and decryption.

- Encryption is the process of translating the plain text data (plaintext) into random and mangled data (called cipher-text).
- Decryption is the reverse process of converting the cipher-text back to plaintext.

3. Explain about proxy server.

- A proxy server acts as an intermediary between the end users and a web server.
- A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resources available from a different server.
- The proxy server examines the request, checks authenticity and grants the request based on that.



Working of Proxy server

4. What are the guidelines to be followed by any computer user?

- **Honesty:** Users should be truthful while using the Internet.
- **Confidentiality:** Users should not share any important information with unauthorized people.
- **Respect:** Each user should respect the privacy of other users.
- **Professionalism:** Each user should maintain professional conduct.
- **Obey The Law:** Users should strictly obey the cyber law in computer usage.
- **Responsibility:** Each user should take ownership and responsibility for their actions.

5. What are ethical issues? Name some.

- An Ethical issue is a problem or issue that requires a person or organization to choose between alternatives that must be evaluated as right (ethical) or wrong (unethical).

Some of the common ethical issues are listed below:

- Cyber crime
- Software Piracy
- Unauthorized Access
- Hacking
- Use of computers to commit fraud
- Sabotage in the form of viruses
- Making false claims using computers

SECTION - D : Explain in detail

1. What are the various crimes happening using computer?

Crime	Function
Cyber Terrorism	Hacking, threats, and blackmailing towards a business or a person.
Cyber stalking	Harassing through online.
Malware	Malicious programs that can perform a variety of functions including stealing, encrypting or deleting sensitive data, altering or hijacking core computing functions and monitoring user's computer activity without their permission.
Denial of service attack	Overloading a system with fake requests so that it cannot serve normal legitimate requests.
Fraud	Manipulating data, for example changing the banking records to transfer money to an unauthorized account.
Harvesting	A person or program collects login and password information from a legitimate user to illegally gain access to others' account(s).
Identity theft	It is a crime where the criminals impersonate individuals, usually for financial gain.
Intellectual property theft	Stealing practical or conceptual information developed by another person or company.
Salami slicing	Stealing tiny amounts of money from each transaction.
Scam	Tricking people into believing something that is not true.

Spam	Distribute unwanted e-mail to a large number of internet users.
Spoofing	It is a malicious practice in which communication is sent from unknown source disguised as a source known to the receiver.

2. What is piracy? Mention the types of piracy? How can it be prevented?

- Software Piracy is about the copyright violation of software created originally by an individual or an institution. Software Piracy is “unauthorized copying of software”.

Types of piracy

1. Duplicating and selling copyrighted programs
 2. Downloading software from illegal network sources.
- An entirely different approach to software piracy is called shareware, acknowledges the futility of trying to stop people from copying software and instead relies on people’s honesty.
 - Shareware publishers encourage users to give copies of programs to friends and colleagues but ask everyone who uses that program regularly to pay a registration fee to the program’s author directly.
 - Unauthorized access is when someone gains access to a website, program, server, service, or other system by breaking into a legitimate user account.
 - To prevent unauthorized access, Firewalls, Intrusion Detection Systems (IDS), Virus and Content Scanners, Patches and Hot fixes are used.

3. Write the different types of cyber attacks.

Virus	A virus is a small piece of computer code that can repeat itself and spreads from one computer to another by attaching itself to another computer file. One of the most common virus is Trojan.
Worms	Worms are self – repeating and do not require a computer program to attach themselves. Worms continually look for vulnerabilities and report back to the author of the worm when weaknesses are discovered.
Spyware	Spyware can be installed on the computer automatically when the attachments are open, by clicking on links or by downloading infected software.
Ransomware	Ransomware is a type of malicious program that demands payment after launching a cyber – attack on a computer system. This type of malware has become increasingly popular among criminals and costs the organizations millions each year.

Chapter-18 Tamil Computing

Very Short Answers

1. List the search engines supported by Tamil language.

- Google and Bing

2. What are the keyboard layouts used in Android?

- Sellinam and Ponmadal are familiar keyboard layouts that works on Android operating system in smart phone using phonetics.

3. Write a short note about Tamil Programming Language.

- Based on Python programming language, the first Tamil programming language “Ezhi” (எழில்) is designed. With the help of this programming language, you can write simple programs in Tamil.

4. What is TSCII?

- TSCII (Tamil Script Code for Information Interchange) is the first coding system to handle our Tamil language in an analysis of an encoding scheme that is easily handled in electronic devices, including non – English computers.
- This encoding scheme was registered in IANA (Internet Assigned Numbers Authority) unit of ICANN.

5. Write a short note on Tamil Virtual Academy.

- Tamil Virtual University was established on 17th February 2001 by the Govt, of Tamil Nadu.
- Now, this organisation is functioning with the name “Tamil Virtual Academy”.
- This organisation offers different courses regarding Tamil language, Culture, heritage etc., from kindergarten to under graduation level.