

H.S.C. BOARD QUESTION BANK OBJECTIVE TYPE QUESTIONS

- 1) Windows NT is ----- operating system. (Oct 2003)
a) Multiuser b) multitasking c) multithreading d) all a,b,c Ans : d) all a,b,c
- 2) ----- is not an operating system (Mar 2004)
a) UNIX b) LINUX c) MS-DOS d) C+++ Ans d) C++
- 3) ----- spread more rapidly but causes less damage to computer network. (Oct 2004)
a) Viruses b) Worms c) Bombs d) none of a,b,c, Ans : b) Worms
- 4) ----- are operating system programs. (Mar 2005)
a) Application programs b) Users programs c) process management d) antivirus programs
Ans c) process management
- 5) ----- is given to each process so that a process does not use the CPU indefinitely. (Oct 2005)
a) Context switching b) Time slice c) Token time d) purchased priority Ans : c) Time slice
- 6) Following is not a process state. (Mar 2006)
a) Ready b) Blocked c) Resumed d) Running Ans : c) Resumed
- 7) In information management ----- service is provided by operating system. (Oct 2006)
a) Change the priority of process b) To allocate a chunk of memory to process
c) Open a file (for read, write or both) d) Wait for a child process to terminate
Ans c) Open a file (for, read, write or both)
- 8) ----- is an operating system. (Mar 2007)
a) C++ b) C c) VB d) LINUX Ans : d) LINUX
- 9) Scheduling policy is the term related to ----- management of operating system. (Oct 2007)
a) Information b) Process c) Device d) Memory Ans : c) Process
- 10) Terminate a process is the system call available in ----- management. (Mar 2008)
a) Process b) Memory c) Information d) File Ans : a) Process
- 11) ----- program is a part of operating system. (Oct 2008)
a) Application b) Antivirus c) Process Management d) User's
Ans : c) Process Management
- 12) The time required for read/write head to move to the correct track is ----- (Mar 2009)
a) Latency time b) Rotational Delay c) Seek time d) none of a,b,c Ans c) Seek time
- 13) Data is instantly updated in case of ----- operating system. (Mar 2010)
a) Batch processing b) Multi Programming c) Multi user d) Real Time Ans d) Real time
- 14) ----- is not operating system programs. (Oct 2010)
a) Process management b) Information management c) Memory management
d) Antivirus program Ans: d) Antivirus program
- 15) A program under execution which completes for the CPU time and other resources is called as -----
a) Process b) multiprogramming c) context switching d) none of a,b,c (Mar 2011)
Ans a) Process
- 16) If the page size for 1MB is 2KB then the number of higher order bit of the address bus used to denote the page number is ----- (Oct 2011)
a) 11 b) 10 c) 9 d) 8 Ans c) 9
- 17) INUX is a ----- software. (Mar 2012)
a) Public domain b) free c) paid d) private Ans b) free
- 18) Terminate a process is system call in ----- (Oct 2012) Ans : b) Process Management
a) Information management b) process management c) memory management d) none a,b,c
- 19) If the page size for 1MB is 1KB then the number of higher order bit of the address bus used to denote the page number is ----- (Mar 2013)
b) 11 b) 10 c) 9 d) 8 Ans b) 10
- 20) Operating system is ----- (Mar 2003, Oct 2002)
a) Hardware b) software c) input device d) output device Ans : b) software
- 21) If the page size for 2MB is 2KB then the number of higher order bit of the address bus used to denote the page number is ----- (Oct 2011)
c) 11 b) 10 c) 9 d) 8 Ans b) 10

Oct 2003

- 1) What is GUI? State any four advantages of GUI. (Mar 2012)
- 2) What is meant by a system call? How it is used? How does an application program use these calls during execution?
- 3) Explain the term multitasking with a suitable example.
- 4) Discuss virus detection, removal and prevention philosophies and methods.
(Oct 2005, Mar 2008, Oct 2011)

Mar 2004

- 5) What is meant by GUI? What are essential components of GUI? Explain any three.
(Oct 2004, Oct 2012)
- 6) What are the components of LINUX operating system? Explain any three features of LINUX.
- 7) What is difference between a worm and a virus? Explain how these can be prevented.

Oct 2004

- 8) What is operating system? Explain the services provided by an operating system. (Oct 2007, Mar 2009)
- 9) Define 'security' with respect to an operating system. Explain the different elements of security.
(Oct 2012)

Mar 2005

- 10) What is GUI? Explain in brief any two features of GUI.
- 11) What is a computer virus? State any four methods by which a virus can infect other programs.
(Oct 2007, Mar 2010, Mar 2011)
- 12) Explain the following elements of data security and protection:
a) Confidentiality b) Integrity c) Availability (Oct 2006, Oct 2008, Oct 2010)

Oct 2005

- 13) Explain the following window operations:
a) Dragging b) Resizing c) Minimizing and Maximizing
- 14) State the different services provided through process management.

Mar 2006

- 15) What is meant by a system call? How it is used? How does an Application Program (AP) use these calls during execution.
- 16) What is computer virus? State the various types and the basis on which they are classified. (Mar 2007)
- 17) Which are the three major areas in which the operating system divides its services? State two system calls of each.

Oct 2006

- 18) Explain any six basic GUI features.
- 19) Explain following terms related to I/O operation:
a) Seek Time b) Rotational Delay c) Transmission Time

Mar 2007

- 20) Explain following with respect to GUI :
a) Scroll bar b) Title bar c) Minimize
- 21) Explain in brief the following programs of MS-Windows: (Mar 2009, Mar 2013)
a) Program Manager b) File Manager c) Control Panel

Mar 2008

- 22) Which are the three major areas in which the operating system divides its services. Give examples.
- 23) Explain in short the functions of menu bar and scroll bar components of GUI.

Oct 2008

- 24) Explain the following components of windows operating system :
a) Task bar b) Minimize button c) Maximize button
- 25) What is computer worm? Explain its mode of operation. (Oct 2010, Mar 2013)

Mar 2009

- 26) Differentiate between computer worms and computer viruses.

Oct 2009

- 27) Explain services of operating system with its system call.
- 28) Explain in short any three ways how security of a system can be attacked for computers in network.
- 29) Explain with respect to process management :
a) Turn around time b) Event response time c) terminal time
- 30) Distinguish between paging and segmentation.
- 31) State features of LINUX.

Oct 2003

- 32) Explain the following terms in case of magnetic disk: (Mar 2006, Mar 2009, Mar 2010)
a) Tracks and Sectors b) Seek Time c) Transmission Time d) Latency/Rotational delay
- 33) With suitable example shows the memory map for a fixed partition allocation with its partition description table.
- 34) Explain the term multitasking with a suitable example.

Mar 2004

- 35) Explain the file system related to Information management with file operation only.(Mar 2007)
- 36) Explain the concept of virtual memory. Explain any three terms used in virtual memory.
- 37) What are the different functions performed by memory management in case of operating system? Draw a memory map of a single user computer. Explain types of partitioning in brief.

Oct 2004

- 38) Explain any four features of WINDOWS 98 operating system.
- 39) What is process? Explain running, ready and blocked process states in process management.(Mar2012, Oct 2012, Mar 2013)
- 40) Explain paging and segmentation. (Mar 2011)

Mar 2005

- 41) Why keyboard is referred as a memory map terminal? Explain the multiple memory locations involved in the input-output operations between the keyboard and screen.
- 42) Explain the single contiguous and fixed partitioned memory management systems with a suitable memory mapping diagram.
- 43) Explain the following terms in case of process scheduling (Oct 2008)
a) Turnaround time b) Waiting time c) Terminal Response time d) Event Response time

Oct 2005

- 44) Discuss how security system can be attacked by different ways.
- 45) What is partitioning? Explain fixed and variable partitioning. (Mar 2009)
- 46) Explain the following with respect to process management:
a) CPU utilization in multiprogramming b) Running state c) Ready state d) Blocked state

Mar 2006

- 47) State various steps involved in allocation of a partition in case of fixed partition memory management. (Mar 2007, Oct 2012)
- 48) With reference to process management explain the following terms : (Mar 2008, Mar 2010, Mar2011)
a) External priority b) Purchased priority c) Internal priority d) Time slice

Oct 2006

- 49) Explain any four features of WINDOWS NT operating system. (Oct 2011, Oct 2012)
- 50) What functions are performed by Memory Management of operating system.? State any four memory management system.
- 51) Explain the following concept of virtual memory : (Mar 2011)
a) Page fault b) Page replacement policy c) Dirty page d) Working set.

Mar 2007

- 52) What are different functions performed by memory management in operating system. Draw a memory map of a single user computer. Explain types of partitions in brief.

Oct 2007

- 53) What is visual display unit (VDU) ? Explain following terms with respect to V.D.U.
a) Dumb terminal b) Intelligent Terminal (Oct 2009)
- 54) Explain in detail Internal and External priority with respect to process scheduling.
- 55) Explain the following terms with reference to virtual memory :
a) Locality of Reference b) Page Fault c) Working Set d) Page Replacement Policy

Mar 2008

- 56) What is partitioning? Explain fixed and variable partitioning in detail.
- 57) Explain use if Video RAM. Explain Data bytes and Attribute bytes.

Oct 2008

- 58) Why video terminal is called memory mapped? Explain the difference between alphanumeric (character oriented) and Graphic (bit oriented) memory mapped video terminal. (Oct 2010)
- 59) Explain Page Memory Management System with a suitable Page Map Table (PMT). (Mar 2010)

Mar 2009

60) What is context switching? Explain context switching at a process level in multiprogramming system with example. (Oct 2010, Oct 2011)

Oct 2009

61) Draw single user memory map. Explain variable partitioning with example.

62) Define GUI. Explain menu bar and scroll bar with diagram.

Mar-20	Explain any three features of windows 98 operating system.
Mar-20	Explain memory map of single user operating system.
Mar-20	Define operating system. In which categories operating system provides it's services?
Mar-20	Explain internal and external fragmentation in memory management of operating system.
Mar-20	What is a virus? Write any three infecting methods of virus.
Mar-20	Explain time sharing related to process management of operating system.
Aug- 18	Define the following terms related to operating system. i) Device Driver ii) Terminal I/O
Aug- 18	Explain three component of MS Windows.
Aug- 18	Explain any four types of security attacks in operating system.
Aug- 18	Write difference between Worm and Virus.
Aug- 18	Explain fixed partitioning and variable partitioning in memory management of operating system.
Aug- 18	Write any six features of Linux Operating System.
Mar- 18	What is File System? Explain tape based and disk based file system.
Mar- 18	What is VDU? Explain following terms of VDU : i) Dumb Terminal ii) Intelligent Terminal
Mar- 18	What is GUI? Explain the following components of GUI : i) Menu Bar ii) Title Bar iii) Scroll Bar
Mar- 18	Explain the following process states : i) Running State ii) Ready State iii) Blocked State
Mar- 18	Define Security. Explain the different elements of security.
Mar- 18	Explain following terms in case of Process Scheduling: i) Turn around Time ii) Waiting Time iii) Terminal Response Time iv) Event Response Time
July -17	What is a Computer Virus? State any four methods by which a virus can effect other programs.
July -17	Write any two important features of the following operating systems: i) Windows 98 ii) LINUX iii) Windows NT
July -17	What is System Call? List any two system calls for Memory Management, Process Management and Information Management.
July -17	What is GUI? State any four advantages of GUI.
July -17	What is partitioning? Explain fixed and variable partitioning.
July -17	Explain the following terms in case of Magnetic Disk: i) Tracks and Sectors ii) Seek Time iii) Transmission time iv) Latency / Rotational Time
Mar -17	Explain any three features of Windows-NT Operating System.
Mar -17	State the various steps involved in the allocation of partition in case of fixed partition memory management.
Mar -17	Discuss Virus Detection, Virus Removal and Prevention Philosophies.
Mar -17	What functions are performed by Memory Management of Operating System? State any four Memory Management System.
Mar -17	With reference to process management, explain the following terms: i) External Priority ii) Purchased Priority iii) Internal Priority iv) Time Slice
Mar -17	What is Computer Virus? What are the different methods by which virus can infect other programs?
July-16	Explain in short the functions of minimize, maximize and close buttons in the MS-windows environment.

July-16	Why video terminal is called memory mapped? Explain the difference between alphanumeric and graphic memory mapped video terminal.
July-16	Explain the functions of the following antivirus programs: i) Virus Detection ii) Virus Removal iii) Virus Prevention
July-16	Explain paged memory management system with a suitable page map table.
July-16	Explain context switching operation between two processes A and B with suitable diagram.
July-16	Explain in brief any three ways the data security system can be attacked and penetrated.
Mar- 16	Explain why user is not allowed to directly interact with the Hard Disk.
Mar- 16	Explain Virus Detection, Removal and Prevention.
Mar- 16	State features of WINDOWS-NT.
Mar- 16	Explain the process of Multiprogramming in Process Management with help of an example.
Mar- 16	What is virtual memory? Explain any three terms related to virtual memory.
Mar- 16	Explain following terms with respect to memory management : i) Internal Priority ii) External Priority iii) Purchase Priority iv) Time Slice
Oct-15	What are the basic services provided by operating system?
Oct-15	Explain the following operations of file management. i) Create a File ii) Writing a File iii) Delete a File
Oct-15	What is Virus? Explain any two infection methods of virus.
Oct-15	What is GUI? Explain any two elements of GUI.
Oct-15	What is multiprogramming and context switching in process management of operating system.
Oct-15	What is paging? Explain using example and diagram of page table, logical memory and physical memory.
Oct-15	Define security of operating system. Describe the elements of security.
Mar-15	What is virtual Memory? Explain any two elements of Virtual Memory.
Mar-15	Give three differences between WORM and VIRUS.
Mar-15	What is System Call? List any two System Calls for Memory Management, Process Management and Information Management.
Mar-15	Explain Multiuser and Time Sharing Operating Systems.
Mar-15	What is File System? List and explain types of File Systems used in OS.
Mar-15	Give features of Windows 98 Operating System.
Oct-14	Give the features of LINUX operating system.
Oct-14	What are the basic services provided by operating system?
Oct-14	Explain the following terms in regard of virtual memory: i) Page Fault ii) Working Set iii) Demand Paging iv) Dirty Page
Oct-14	Explain contiguous allocation method in information management. Give any two advantages of it.
Oct-14	List and explain the types of memory partitioning use in memory management of operating system.
Mar-14	Explain three process states of Operating System.
Mar-14	State functions of Memory Management.
Mar-14	Define a Block of Data. State necessary parameters to be provided to O.S. to accesses a Block of Data.
Mar-14	Distinguish between Worm and Virus.
Mar-14	State features of LINUX Operating System.
Oct-13	Explain any two important features for each of the following operating system : i) Windows 98 ii) Windows NT iii) LINUX
Oct-13	What is Video RAM? Explain the use of attribute byte and data byte of Video RAM with example.
Oct-13	Explain the function of the following Antivirus Programs: i) Virus Detection ii) Virus Removal iii) Virus Prevention
Oct-13	Explain in short the three main function to be carried out by Memory Management module of an operating system. State any two types of Contiguous Real Memory Management System.

Oct-13	What are computer worms? Explain its modes of operation.
Oct-13	Explain the term multitasking with suitable example.
Mar-13	Explain in short the function of following MS-Windows environment : i) Program Manager ii) File Manager iii) Control Panel
Mar-13	What is a Video RAM? Explain use of attribute byte and data byte of video RAM with example.
Mar-13	What is a computer worm? Explain its modes of operation.
Mar-13	Explain in short the three main functions to be carried out by Memory Management Module of Operating System. State any two types of Contiguous Real Memory Management System.
Mar-13	Explain in brief three major services provided by the operating system.
Mar-13	What is process? Explain running, ready and blocked process states of Process Management.
Oct-12	What is GUI? Explain any two essential components of GUI.
Oct-12	Explain any four features of Window-NT operating system.
Oct-12	Define the term 'Security' with respect to an operating system. Explain any two elements of security.
Oct-12	Define the term Process. Explain running, ready and blocked process state in a Process Management.
Oct-12	State the different services provided through Information Management Service of Operating System.
Oct-12	State various steps involved in the allocation of a partition in case fixed partition memory management.
Mar-12	What is GUI? State any four advantages of GUI.
Mar-12	What are different functions performed by Memory Management in an Operating System? Explain types of partition in brief.
Mar-12	What is the difference between a worm and virus? Explain how these can be prevented?
Mar-12	Define a block with reference to Operating System. What are the parameters of a block of data that are concerned with an Operating System. Explain in brief.
Mar-12	What is a Process? Explain ready, running and blocked process states in Process Management.
Oct-11	Explain the use of video RAM. Explain data byte and attribute byte.
Oct-11	Explain any four features of WINDOWS-NT Operating System.
Oct-11	Discuss virus detection, removal and prevention philosophies and methods.
Oct-11	Differentiate between paging and segmentation.
Oct-11	Explain context switching at a process level in Multi Programming System with example.
Mar-11	Explain in short the function of Menu bar and Scroll bar components of GUI
Mar-11	Explain the following terms: i) Page Fault ii) Working Set iii) Page Replacement Policy iv) Dirty Page
Mar-11	What is Computer Virus? State any four methods by which Virus can infect other programs.
Mar-11	With reference to Process Management, explain the following terms: i) External Priority ii) Purchased Priority iii) Internal Priority iv) Time Slice
Mar-11	Explain with the help of memory map diagram of single user operating System.
Mar-11	Explain Paging and Segmentation.
Oct-10	In case of Windows Operating System, explain the following options of File Menu: i) Open ii) Print iii) Save as
Oct-10	Why video terminal is called as memory mapped? Explain difference between alphanumeric (character oriented) and graphic (bit oriented) memory mapped video terminals.
Oct-10	What is Computer Worm? Explain its mode of operation.
Oct-10	Explain Single Contiguous and Fix Partition Memory Management System with suitable memory mapping diagram.
Oct-10	Explain the following elements of Data Security: i) Availability ii) Integrity iii) Confidentiality
Oct-10	Explain context switching at a process level in multi-programming with a suitable example.
Mar-10	Explain in brief : i) Device Driver ii) Types of Memory Mapped I/O Terminals

Mar-10	Explain the following terms in case of Magnetic Disk: i) Tracks and Sectors ii) Seek Time iii) Transmission Time iv) Latency/Rotational Time
Mar-10	What is a Computer Virus? State any four methods by which a virus can affect other programs.
Mar-10	Explain Paged Memory Management System with a suitable Page Map Table (PMT).
Mar-10	Explain in brief any three ways by which a system can be attacked and penetrated
Mar-10	With reference to Process Management explain the terms : i) Internal Priority ii) Time Slice iii) External Priority iv) Purchase Priority

H.S.C. board Probable Marks : 21

1 Mark	Objective	1	1X1	= 01
3 Mark	Question	4	4X3	= 12
4 Mark	Question	2	2X4	= 08

Q) What is operating system ? Write its function.

An Operating system is a system program which acts as an interface between the user of computer and the computer hardware.

Functions performed by operating system : -

The following functions are performed by an operating system :

- 1) The primary aim of the operating system is to make the hardware convenient to use.
- 2) Operating system helps to user to executes program.
- 3) Operating system controls execution of program to prevent errors and improper use of computer system.
- 4) Operating system makes provision for security of information to users.
- 5) Operating system eliminate duplicate errors by number of programmers in development of complicated routines.
- 6) Operating system provides facility to shares the same hardware among the users.
- 7) Operating system schedules resources (h/w or peripherals) among users.

Q) What are the features of windows 98?

The main features of windows 98 are as listed below:

i) Easier to use :-

- 1) Windows 98 is a single user multitasking operating system.
- 2) Navigating around the computer is easier in windows 98.
- 3) A file can be opened by a single click.
- 4) Windows 98 allows us to use multiple monitors with single computer.
- 5) New hardware can be easily installed and used without restarting computer.
- 6) With windows 98 we can use digital cameras and other digital imaging devices.

ii) Faster

- 1) Windows and programs open faster than in windows 95.
- 2) The computer speed and efficiency can be easily improved by simple maintenance.

iii) True web integration:-

- 1) Windows 98 can be easily connected to internet.
- 2) Web pages can be viewed in any window.
- 3) Using Microsoft Outlook Express, Emails and messages can be send to internet newsgroups.
- 4) The internet conferences can also be arranged.

iv) More entertaining :-

- 1) Windows 98 support DVD and digital audio. User can play high quality digital movies and audio on the computer.
- 2) The television broadcast can also be seen.

Q) What are features of Windows NT?

OR Explain any four features of Windows NT Operating System.

The features of Windows NT are as listed below:

- 1) Windows NT is multitasking, multiuser and multithreading operating system.
- 2) A user will get faster response eventhough multiple applications are running.
- 3) Windows NT supports virtual memory management system to allow multiprogramming.
- 4) Symmetric multiprocessing in windows NT allows it to schedule various tasks on any CPU in a multiprocessor system.
- 5) Windows NT is a 32-bit operating system.
- 6) Windows NT uses NTFS (New Technology File system) which implements fault tolerance, security and has support for very large files,

Q) What are features of Linux operating system?

Some of the features of Linux are as given below :

- 1) Linux is a multiuser and extension of unix operating compatible tools.
- 2) Linux runs on a wide variety of platforms. It was developed exclusively on PC architecture.
- 3) Linux provides as much as functionality from limited resources. It can runs on machine having 4MB of RAM.
- 4) Linux present standard interfaces to both the programmer and user.
- 5) Linux supports a wide base of applications.
- 6) Linux is free software that means people can copy it, modify if and use it in any manner they want.
- 7) The file system in Linux obeys Unix semantics.
- 8) The memory management system uses page sharing and copy-on-write to minimize duplication of data shared by different processes. Pages are loaded on demand when they are referenced first.

Q) Which are the three areas in which the operating system divides its services? Give examples of each.

The operating system divides its services in the following three main areas :

- 1) Information Management (IM) 2) Process Management (PM) 3) Memory Management (MM)

1) Information Management (IM) :-

Information Management (IM) provides facilities like store, retrieve, modify etc on various devices.

Services of information management :-

- i) Create files or directories.
- ii) Opens files or explore directories.
- iii) Delete, copy and close files.
- iv) Change working directory.

2) Process Management (PM) :-

The process Management (PM) provides facilities like keeps track of all running programs which called as processes. In multiuser operating system number of users located at different programs at a time. In such case operating system keeps track of all processes. It schedules them and dispatches them.

Services of Process Management (PM) :-

- i) Read a process.
- ii) Block a process.
- iii) Resume a process.
- iv) Terminate a process.
- v) Suspend a process.
- vi) Delay a process.
- vii) Change priority of process.
- viii) Generate a process.

3) Memory Management (MM) :-

The Memory Management (MM) provides facilities like allocation and deallocation of memory to process and from process.

Services of Memory Management (MM):-

- i) To allocate a chunk of memory to process.
- ii) To free chunk of memory from a process.

Q) Explain the following terms in case of magnetic disk:

- a) Track and Sector b) Seek Time c) Transmission time d) Latency time/ Rotational delay

a) Track and Sector :-

Magnetic disk i.e hard disk and mass storage surface is made up of concentric circles called as tracks. The number of tracks varies depending on the disk type. A track is further divided into smaller areas called as sectors i.e they are circular region of data on platter.

A sector is a smallest unit of information which can be read from or written to the disk. Sector varies from 32 bytes to 4096 bytes and track contains 4 to 32 sectors per track and from 75 to 500 tracks per disk surface.

b) Seek Time:-

The time required for read/write heads to moves to the correct track is called as seek time.

c) Transmission Time:-

The time required for active read/write head for appropriate surface and read data is called as transmission time.

d) Latency Time OR Rotational Delay:-

The time required for requested sector on track to rotate below the head is called as latency time or rotational delay. The time is calculated as : $\text{Time} = \text{one revolution} / 2.$

Q) What is meant by a system call? How it is used. How does an application program (AP) uses these system calls during execution.

System Call :- System call provides the interface between a process and the operating system.

System calls are used in different ways as below:-

a) Some system may allow system calls to be made directly from a higher level language program. In this calls are predefined function or subroutine. b) Some languages like C, PERL allows system calls to be made directly. c) Fortran language provides set of library routines.

Application program uses system calls during execution as below

An application program uses sequence of system calls during execution such like :

- a) To prompt a message on terminal application program uses system calls.
- b) To read from terminal application program uses system calls.
- c) To each and every task program application program uses system calls.

Q) Enlist system calls of information, process and memory management of operating system

1) System calls of Information Management (IM) :-

- i) Create files or directories. ii) Opens files or explore directories. iii) Delete, copy and close files.
- iv) Change working directory. v) Close file. vi) Read data from file to buffer.
- vii) Write data from buffer to file. viii) Move file pointer. ix) Create a link. x) change working directory.

2) System calls of Process Management (PM) :-

- i) Read a process. ii) Block a process. iii) Resume a process. iv) Terminate a process. v) Suspend a process.
- vi) Delay a process. vii) Change priority of process. viii) Generate a process.

3) System calls of Memory Management (MM) :-

- i) To allocate a chunk of memory to process. ii) To free chunk of memory from a process.

Q) What is device directory?

The disk has a device directory indicating which files are on the disk. The device directory is stored at fixed address such as 00001. The list of information which is kept in directory entry is given below:

1) File name : The symbolic file name. **2) File Type:** The type of file for those systems which supports different types. **3) Location:** A pointer to device and location on that device of the file.

4) Size: The current size of files bytes, words, or blocks and maximum allowed size.

5) Current position: A pointer to current read or write position in the file.

6) Protection: Access control information to control reading, writing, executing and so on.

7) Usage count: Indicating the number of processes which are currently using have opened the file.

8) Time, date and process identification: This information is kept such as creation , last modification and Last use. Thus it is useful for protection and usage monitoring. It may take from 16 to 1000 bytes to record this information for each file.

The directory can be organized in any ways such as linear list and hash table.

Q) What are the components of Linux system?

The Linux operating system is composed of three main bodies or sections as like below:

1) Linux Kernel :-

Kernel of Linux maintains all important abstractions of the operating system such as processes and virtual memory.

2) Linux system shared libraries:-

The system shared libraries of Linux operating system defines a standard set of functions through which applications can interact with the kernel and which implements much of the operating system functionality.

3) System Utilities of Linux:-

System utilities are system software programs compilers, loader, linkers etc.

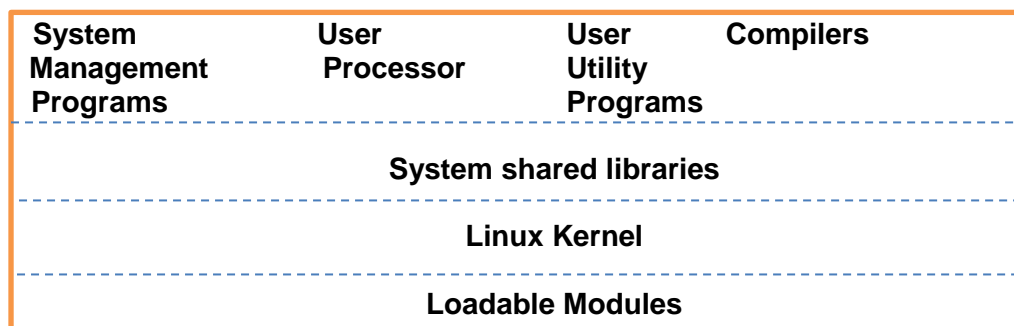


Fig Various components of Linux Operating System

Q) What is file system? What are two types of file system?

The collection of related information i.e. processed data or programs is called as file. Each file has specific name which is used to refer that file. For convenient use of the computer system the operating system provides a uniform, logical view of information storage. The operating system manages mass storage devices such as hard disk, pendrives, CD-ROM, memory card etc to implement the abstract concept of file.

The operating system maps files onto physical devices such as magnetic tapes or hard disk. The Information Management of an operating system allows user to define files and directories and allocates or deallocates the disk space to each file.

Types of file system:-

There are two types of file system as like below :

1) Tape-based File system :-

These are serial or sequential access storage device. In this files are stored onto reels of physical tapes. These are not sufficient to use because with slower access mechanism and sequential access. The R/W process are time consuming method.. In banking these are used for to backup files.

2) Disk- based File system :-

These are direct access storage medium. The R/W operations in this medium is very high speed it is equal to time of input data. It is sufficient to use as storage medium in computer system to store files.

Advantages of disk-based systems over tape-based system:-

The following are advantages provided by disk based system:

- 1) Finding a file on tape based system is difficult and time consuming while a file can easily be found on disk based system.
- 2) In tape-based system if a file is to be modified it requires to copy entire tape whereas in a disk based system a file can be modified easily without copying the entire disk.
- 3) The disk-based have large storage capacity to store files than tape based system.
- 4) Disk based system are easier and convenient to use instead of tape based system.
- 5) Each disk consists of number of blocks which can be rewritten easily while we require entire tape to copy if we have to rewrite something.

Functions of File System : The file system of an information management performs following different functions

- 1) It allows the user to define files and directories and allocate/deallocate the disk space to each file.
- 2) It stores data or information.
- 3) It arranges various files of same type under one directory.
- 4) It opens file, create a file , delete a directory and set certain access controls on a file.

Q) Explain the file system related to Information Management with file operations only.

The file system related to IM allows the user to define files and directories and allocates or deallocates the disk space to each file. A file is a collection related information. It can program or data.

File Operation in Information Management:-

The file concept is implemented by the operating system. An IM provides system calls such as create, read, write, rewind and delete files etc.

- 1) **Create a file :-** For creating a file first whether sufficient space is available for that file is checked in disk. If it is available then entry for new file must be made in directory.
- 2) **Write to a file:-** For writing to a file first there is command in which name of the file is given. Then operating system searches for that file in directory entry and write to it.
- 3) **Reading a file:-** For reading a file there is a system call in which file name is specified. Then operating system searches for that file in directory entry and read it.
- 4) **Rewind a file:-** The directory is searched for appropriate entry and file is reset to the beginning of file.
- 5) **Delete a file :-** To delete a file directory entry is searched and if file is found it releases the memory space and that directory entry now become invalid.

Q) What is access method? Explain different access methods.

File stores information. When it is used then this information must be accessed and read into computer memory.

Types of access methods:

There are several methods to access the information in the file

- 1) Sequential Access Method : Here file is accessed sequentially. In this a program skip forward or backward 'n' records. In this operations such as read and write.
- 2) Direct Access Method : Direct access is based upon a disk model of a file. Here the file viewed as a sequence of blocks or records. In this method any block can be read or written.

Q) What is fragmentation? Explain types of fragmentation in information management.

OR What is fragmentation? Explain Internal and External fragmentation.

Definition of Fragmentation:-

The method of dividing or partitioning a main memory space or random access memory is called as fragmentation. They are of two types :

1) Internal Fragmentation:-

Wastage of memory space within partition is called as internal fragmentation. A file consists of number of blocks. Consider the block size of operating system is 1024 bytes and a file of 3499 bytes. Then when it is loaded in memory for execution it would have allocated 4 blocks. Thus last 597 bytes would be wasted. This is called as internal fragmentation. Thus larger block size causes more internal fragmentation.

2) External Fragmentation:-

Variable partition suffers from external fragmentation. Suppose a job of 512 bytes is terminated and new job is 256 bytes is loaded in the partition then 256 bytes of memory is wasted. This wastage is called as external fragmentation.

Q) What are device driver?

The device drivers are system software programs it is required for each input and output devices as well as storage devices. Thus each device will require different drivers as per functionality. A device driver knows how the buffers, flags, register control and status bit should be used for a particular device. Some device drivers are useful for data conversion. For simply reading a character from a device involves complex sequence of device specification operation. Rather than writing the code every time the device driver was simply used from library routine.

Functions of device management or device driver :

The device driver performs different types of functions as follows:

- 1) It written special subroutine for each I/O device such subroutine is called device driver.
- 2) It moves data from deriver to terminal.
- 3) Block numbers are converted into sector number.

Q) What is V.D.U. ? Explain Dumb Terminal and Intelligent Terminal.

OR What is terminal ? Explain two types of V.D.U. terminal.

A visual display unit (V.D.U.) or terminal is nothing but a common Input/Output medium. There are generally two parts of terminal I/O hardware such as below:

- 1) Dumb Terminal
- 2) Intelligent Terminal

1) Dumb Terminal:-

A dumb terminal is consist of a microprocessor on which small programs can runs on limited amount of memory. It is responsible for basic Input/Output operations. A dumb terminal does no processing on input character.

2) Intelligent Terminal:-

An intelligent terminal has powerful hardware and software than dumb terminal. It also has more amount of memory. It can carry out some process on input characters also it can process programs independently.

Q) Explain the use of video RAM. Explain data bytes and attribute bytes with diagram.

The video RAM is basically the memory that the terminal hardware itself has. Anytime all characters stored in the video RAM are displayed on the screen by video controller using display electronics. All particular information i.e. attributes are stored in video RAM. Video RAM consists of 2000 databytes it is stored in memory from 0 To 1999 preceded by 2000 corresponding attribute bytes they also stored in memory from 0 To 1999 as shown in following figure

Attribute Byte 0	Data Byte 0	Attribute Byte 1	Data Byte 1				
						Attribute Byte 1999	Data Byte 1999

A typical alphanumeric screen can display 25 lines each consisting of 80 characters i.e. $25 \times 80 = 2000$ characters.

1) Data Byte

All 2000 characters are stored in video RAM. To display any specific character on the screen at a specific position all ASCII or EBCDIC code for characters is to move in video RAM.

2) Attribute Byte

There is one attribute byte for each data byte. This byte tells the video controller how the character is to be displayed. It signifies whether the corresponding data character which is stored next to it in the video RAM is to be displayed bold, underlined, blinking or in reverse video etc

Q) Why keyboard is referred as memory map terminal? Explain the multiple memory location involved in the input-output operations between the keyboard and screen.

Terminals have a video RAM is generally with 2000 data bytes preceded by 2000 corresponding attribute bytes. Anytime all the 2000 characters i.e. 25 lines X 80 columns on screen are stored in video RAM are displayed on the screen by the video controller.

The video RAM is treated as part of the main memory only. Thus for moving any data in or out of the video RAM ordinary load or store instructions are sufficient. So keyboard is referred as memory map terminal.

The following multiple memory locations are involved in the input-output operations between the keyboard and screen :

1) Small memory within the keyboard itself

When character is keyboard in the 8-bit ASCII or EBCDIC code is generated which is stored temporarily in the memory of the terminal itself.

2) The video RAM(data byte and attribute byte)

The ASCII or EBCDIC code for the character is to move to the video RAM at the corresponding position with appropriate coordinates.

3) The operating system buffers

The operating system has one buffer for each terminal and two separate buffers for input and output operations.

4) The Input/Output area of the application program

When the user finishes keying in the data then data is stored in operating system buffer for that terminal is flushed out to the input/output area of the application program which wants that data.

Q) What is a process?

- 1) Process is a task.
- 2) Process is an asynchronous activity.
- 3) Process is an animated spirit of procedure.
- 4) Process is the locus of control of procedure in execution.
- 5) Process is defined as a program under execution which competes for CPU time and other resources.

Q) Defines the terms: 1) Context switching 2) Degree of multiprogramming**1) Context Switching :-**

In multiprogramming system a multiple processes are running at the same time such that when process 1 wait for an I/O and process 2 executes and vice versa. Then lost in time in turning the attention of CPU from one process to another is called as context switching.

2) Degree of multiprogramming:

The number of processes are running simultaneously and hence competing for CPU is known as degree of multiprogramming.

Q) Explain context switching at a process level in multiprogramming system with example.

Multiprogramming is the concept of increasing utilization of CPU by always having something for CPU to executes. In multiprogramming CPU can executes two or more processes simultaneously when process 1 waits for an external event such as an I/O operation then CPU executes process 2 and vice versa. When CPU switches from one process to another then time required for switching is called as context switching.

Let A and B the two processes ready for execution and requires CPU time for execution. Let CPU time be given to process A which is having some instructions depending on process B or on some external event such as an I/O operation. Then it is the job of operating system to halt the execution of process A and give CPU time for process B.

Thus the loss in time in turning the attention of CPU form process A to process B is known as context switching. During context switching the status of CPU registers and flags of the old process are stored in memory area is called as register save area and then CPU starts executing another process. Both processes are in main memory even after context switch.

e.g.

Q) Explain Running, Ready and Blocked process states in process management.**OR Discuss various process states with examples.**

In order to manages switching between processes the operating system defines three basic process states they are such as below:

1) Running State:

It is executed on a processor. There is only process which is executed by CPU at any given moment. This process is called as running process.

In multiprocessor system with multiple CPU there are many running processes at a given moment. The operating system keeps track of all of them.

2) Ready State :

It could be executed on processor if one were available. The process which is not waiting for an external event such as I/O operation but which is not running is said to be in ready state.

Actually a process in ready state could have been running. But the fact is that there is only one CPU which is executing some other process while this process is waiting for CPU's attention towards it.

3) Blocked State :

It is waiting for some event from I/O. When a process is waiting for an external event such as an I/O operation the process is said to be in blocked state.

Q) Write a difference between blocked and ready process.

The major difference between blocked and ready process is that a blocked process can not be directly scheduled even if CPU is free whereas a ready process can be scheduled if the CPU is free.

Q) What is process scheduling? Explain scheduling objectives.

In multiuser operating system a number of programs are running simultaneously and these are called as processes. In this case the operating system keep track of all these processes and will have to dispatch them one after another. This is known as process scheduling.

Objectives of process scheduling:

While scheduling various processes there are many objectives for the operating system. Some of these objectives conflict with each other. Therefore the operating system designers have to choose a set of objectives to be achieved. Some of these objectives are as listed below :

- 1) Fairness:** Fairness refers to being fair to every user in terms of CPU time.
- 2) Throughput** Throughput refers to the total productive work done by all the users put together.
- 3) CPU utilization** CPU utilization is the fraction of the time that the CPU is busy.
- 4) Turnaround time** Turnaround time is elapsed time between the time a program or a job is submitted and the time when it is completed.
- 5) Waiting time** Waiting time is the time a job spends waiting in the queue of the newly admitted processes for the operating system to allocate resources to it before commencing its execution.
- 6) Terminal Response time** In the time sharing system terminal response time is the time to respond with an answer or result to a question and it depends on degree of multiprogramming the efficiency of hardware with operating system and policy of operating system to allocate resources.
- 7) Event response time** In real time system event response time is the time to respond with an event.

Q) What are preemptive and non preemptive philosophies of scheduling.

There are basically two scheduling philosophies depending upon the need. The operating system designers has to select one of them.

Preemptive Philosophy:

This philosophy allows a higher priority process to replaces a currently running process even if its time slice is not over or it has not requested for any external operation. This requires context switching more frequently. It is suited for online real time processing where interactive users and high priority processes requires immediate attention. Thus preemptive philosophy increases fairness of the system but it decreases throughput.

Non preemptive philosophy:

In a non preemptive philosophy a running processes retains the control of the CPU and all the allocated resources until it surrenders control to the operating system on its own wish. This means that even a high priority process enters the system then process can not be forced to give up control. However if the running processes is blocked then due to some external request another process can be scheduled. It is not suited for real time systems where high priority events requires an immediate attention. Thus non preemptive philosophy increases throughput of the system but it decreases fairness.

Q) What is time sharing ?

The concept of time sharing was demonstrated as early as 1960. They provide interactive use of computer. It uses CPU scheduling and multiprogramming to provide each user with a small portion of time-shared computer. Each user has a separate program in memory. When program executed then it executes for only short time before it either finishes or needs to perform I/O.

The operating system will rapidly switch the CPU to the program of some other user. A time shared operating system allows many users to simultaneously share the computer. Each action or command in time shared system tends to be short only a little CPU time is needed for each user. As the system rapidly switches from one actually one computer is shared among many users.

Q) What is priority? Explain internal and external priorities.

The concept of arranging ready processes in a queue so that they can be dispatched one after another for execution depending on some policy is known as priority. Due to many processes competing for the same available resources like CPU and memory concept of priority is used.

Types of priority: There are two types of priorities such as

1) External OR Global priority :

An external priority is specified by the user externally generally at the time of initiating the process. In many cases the operating system allows user to change its priority externally even during its execution. If the user does not specify any external priority at all the operating system then it assumes a certain priority called as the default priority. But when an urgent job needs to be done the system manager permits the process to be created with a higher priority.

2) Internal OR Local priority:

An internal priority is used by scheduling algorithm this based on their calculations on the current state of the process

e.g. user while firing a process can be forced to specify the expected time that the process is likely to take for completion. The operating system can then set internal priority which is highest for the shortest job (SJF : shortest job first algorithm).

Advantages of Internal priority :

Internal priority has two advantages as stated below:

- i) If short jobs are finished faster then the number of processes competing for CPU will be decreased.
- ii) The number of satisfied users will increase.

Q) Define purchase priority and time slice.

1) Purchase priority:

Purchase priority is used in some data centre situations where each user pays for the time used. Higher priority processes are charged at a higher rate to prevent each user from firing his job at the highest priority. This is known as scheme of purchased priority. An Operating system keeps track of the time used by each processes and the priority at which it was used.

2) Time slice :

Each process is normally given certain time to run irrespective of its importance. This is known as time slice. Time slice is given to each process so that a process does not use the CPU indefinitely.

Q) What is multitasking with suitable example? Write advantages of multitasking.

A task can be defined as an asynchronous code path within a process. In multitasking a process can consists of tasks which runs simultaneously. Multiple tasks should be able to runs concurrently within a processes. Multitasking allows programmer flexibility and improves CPU utilization. It reduces the overhead if switching at a process level.

Example: Consider a process consisting of two tasks as follows:

```
Task 0 :      Read a Record
              Process a Record
              Task 0 end.
Task 1:      Write a Record
              Task1 end
```

In above example two tasks are defined within the same process. They run concurrently within the same process it synchronized properly.

If task 0 is blocked then instead of blocking entire process the operating system will find out whether task 1 can be scheduled. When both tasks are blocked then only entire process is blocked. Again if one task is ready then the process can be moved to ready list and then scheduled.

Advantages of multitasking operating system:

It provides following different types of operating system as below:

- 1) Multitasking operating system allows programmer flexibility and also improves CPU utilization.
- 2) When various tasks are defined in a process then process would be blocked only if all the tasks in that process are blocked.
- 3) Again even if any task becomes ready then process can be moved to ready lists from blocked lists.
- 4) By adding task levels context switching at various process levels can be reduced.
- 5) Multitasking is less time consuming and it reduces turnaround time.

Q) What are functions are performed by memory management of operating system? State different types of memory management systems.

Functions performed by Memory Management

Memory management of operating system performs following functions

- 1) MM keeps track of all memory locations free or allocated and if allocated to which process and how much.
- 2) MM decides memory allocation policies such as which process should get how much memory when and where.
- 3) MM uses various techniques and algorithm which allocates and deallocates memory locations. Normally this is achieved with the help of some special hardware.

Different types of Memory Management System:

The memory management of operating system are classified as below

- 1) Contiguous and Real Memory Management System
 - a) Single contiguous b) Fixed partition c) Variable partition
- 2) Non contiguous and Real Memory Management System
 - a) Paging b) Segmentation c) Combined
- 3) Non contiguous and Virtual Memory Management System : Virtual Memory

Q) Explain memory map of single user operating system.

Fig Memory map of single user os

Free Memory
Process
Command Interpreter
Kernel

The Disk operating system (DOS) is called as Single user single tasking operating system. This operating system has memory map of various blocks as like below

1) Free Memory

The program or process is executed if free memory is available.

2) Process The program under execution is called as process.

3) Command Interpreter : It is invoked when the computer is started. The command interpreter sets instruction pointer and executes program. Thus it interprets meaning of command and decides the nature of execution.

4) Kernel: The kernel of OS provides basic operating systems services or operations while the command interpreter interprets the command.

Q) Explain the single contiguous memory management systems with a suitable memory mapping diagram.

Operating System (Monitor)
User Process Area

In single contiguous memory management system the physical memory is divided into two contiguous areas. Such as one area is permanently allocated to the resident portion of the operating system and the other area is used for user process.

As shown in figure operating system may be loaded at lower addresses i.e. 0 TO P. At any time only one user process is in the memory. This process is run to completion and then the next process is brought in the memory.

Q) Difference between contiguous memory management and Non contiguous memory management

Contiguous Memory Management	Non Contiguous Memory Management
1) This system expect the program to be loaded in contiguous memory location.	1) In this system program is divided into different chunks and to be loaded at different portions of the memory.
2) Examples: i) Single contiguous memory management system. ii) Fixed partition memory management system.	2) Examples : i) Paged memory management system. ii) Segmented memory management system. iii) Combined memory management system.

Q) What is partitioning? Explain different types of partitioning methods.

OR What is partitioning? Explain fixed and variable partitioning.

Partitioning means dividing main memory into various sections. These sections are called as partitions. The partitioning is used by operating system by MM to allows multiprogramming.

Types of partitions :

There are two types of partitioning methods such as :

- 1) Fixed partitioning OR Static partitioning
- 2) Variable partitioning OR Dynamic partitioning

1) Fixed Partitioning OR Static Partitioning:

In this partitioning the size is fixed or constant. The partitioned segments are different sizes and fixed. But once they are decided then can not be changed. In this method partitions are fixed at the time of system generation. At this time system manager has to declare the partition time. Fixed partition is also called as static partition. On declaring fixed partitions the operating system creates a partition description table (PDT).

Disadvantages of fixed partitioning:

- 1) Fixed partitioning suffers from internal fragmentation i.e. wastage of memory space within the partition. **e.g.** suppose the partition size is 200KB and a job is of 100KB then 100KB of memory will be wasted. This wastage of memory is called as internal fragmentation.
- 2) Fixed partitioning reduces degree of multiprogramming.
- 3) Fixed partitioning restricts CPU utilization.

2) Variable Partitioning OR Dynamic Partitioning :

The fixed partition suffers form the problem of internal fragmentation i.e wastage of memory space within partition. It also limits the degree of multiprogramming and CPU utilization. The variable partition overcomes these problems.

In variable partitioning number of partitions and their sizes are variable. They can not defined at the time of system generation. These partitions are created by operating system at the run time and they are different sizes. In variable partitioning starting address of partition is not fixed.

Difference between Fixed and Variable partition

Fixed Partition	Variable Partition
1) Partition created could not be changed.	1) Partition created can be changed.
2) Partition can be defined at the time of generation.	2) Partition can not be defined at the time of generation.
3) If suffers form problem of internal fragmentation.	3) Problem of internal fragmentation is solved.
4) Number of partition and their sizes are fixed.	4) Number of partition and their sizes are variable.

Q) Define a block with reference to operating system. What are the parameters of a block of data that are concerned with an operating system. Explain in brief.

Block : A block is a logical unit of data that operating system defines for its convenience. Thus block is a contiguous set of bits or bytes that forms an identifiable unit of data.

Parameters of Block: An operating system has following parameters of a block:

- 1) **File ID :** It is a letter code given to each type of data file to make it easier for the operating system to identify the files.
- 2) **The starting position in the file:** It is the address of memory location from where file will start.
- 3) **The number of bytes to be read:** It is total size of file in bytes.
- 4) **The starting address of memory where data is to be read:** It is starting address of block. Whenever program read any data the file system translates request into reading one or more sector from disk and instruct device driver to read these sector. Operating system keeps all its data structure in terms of block. O.S. translates a block number into sector number. File system request to read desired blocks. It uses disk space allocation i.e. linked list to carry out translation. File system then request deice driver to read desired block. Device driver issues instruction to the controller for the disk to read required blocks. Controller reads data sector by sector by sector and stores it in its own memory until desired block are read in.

Q)State the various steps involved in the allocation of a partition in case of fixed partition memory management.

When a process is to be allocated a partition following take place

- 1) The long term process scheduler of the process management decides which process to be brought into the memory.
- 2) Then it find out the size of the program to be loaded by consulting the information management portion of the operating system. For this the compiler keeps track of size of the program in the header of the executable file.
- 3) Then it makes a request to the partition allocation routine of the memory management to allocates a free partition with the appropriate size. For this partition description table is useful.
- 4) With the help of the information management module it now loads the binary program in the allocated partition. It then makes an entry of the partition ID in the PCB (process control block) before the PCB is linked to the chain of ready processes by using the process management module of the operating system. Thus the routine in the memory management now redefines partition description table (PDT) and marks the status of that partition as allocated called ALLC. The process management eventually schedules this process.

Consider the following partition description table (PDT)

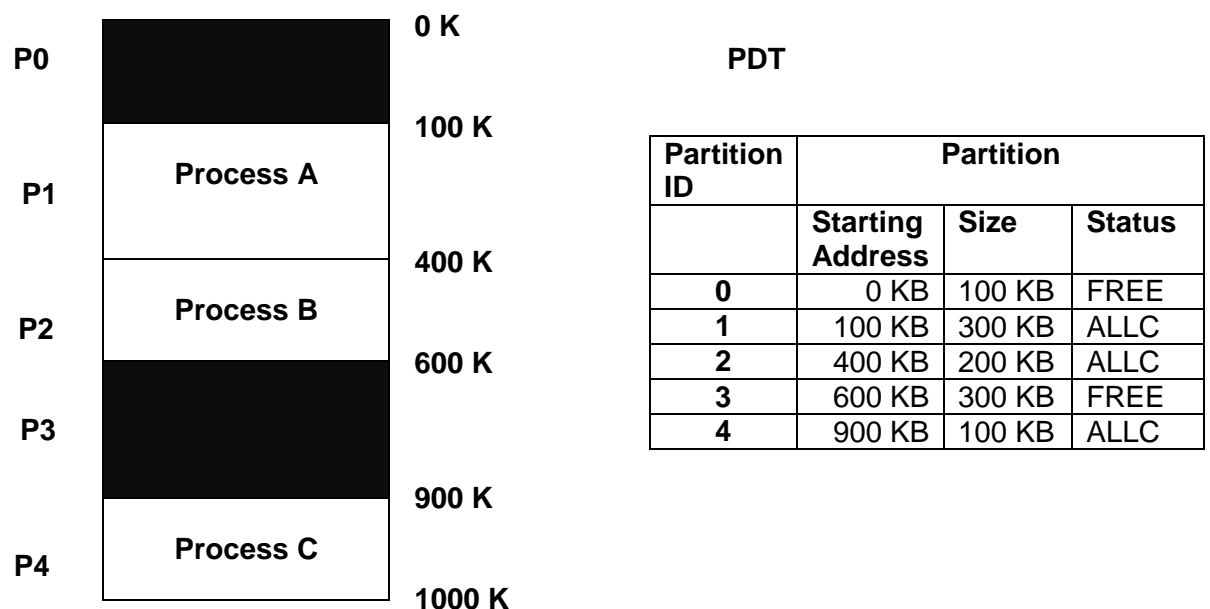


Fig Fixed Partition

Q) Differentiate between Paging and Segmentation.

Paging	Segmentation
1) Pages are physical in nature.	1) Segments are logical division of programs.
2) Pages are of fixed size.	2) Segments are of variable size.
3)Page table translate from user pages to memory frames.	3)The Segment table has a separate entry for each segment in physical memory and length of that segment.
4) In paging logical memory is divided into physical memory.	4) In segmentation main program is divided into segments.

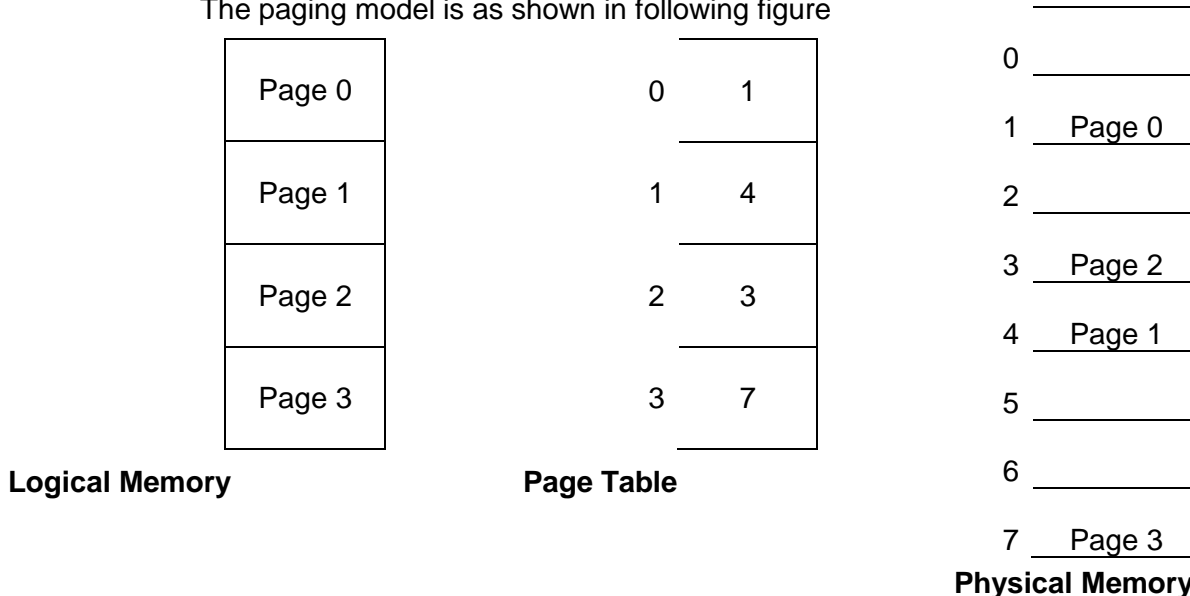
Q) What is paging? Explain in detail.

OR Explain page memory management system with a suitable page map table (PMT)

The fixed partitions suffers from internal fragmentation while variable partition is suffers from external fragmentation because of available memory is not contiguous.

The logical or virtual address space of program is divided into equal sized pages and the physical main memory is also divided into equal sized page frames. The size of page is the same as that of page frame so that a page can exactly fit into a page frame and therefore it can be assigned to any page frame which is free.

The paging model is as shown in following figure



When a computer program is to be executed then its pages are loaded into any available frames and page table is defined to translate from user pages to memory frames. The page size is defined by hardware. It is typically a power of 2.

Q) Explain various disk space allocation with their merits and demerits.

There are two major types for the allocation of disk space to files these are :

1) Contiguous Allocation: It requires each file to occupies a set of contiguous address on the disk. An unallocated segment is called as hole. If a new file is created and there are set of holes then following strategies is used and selects a free hole from set of holes.

i)First Fit: This allows first hole that is big enough. **ii)Best Fit :**This allocate the smallest hole that is big enough. **iii) Worst Fit:**This allocates largest hole.

Merits : If processing is sequential and if the operating system uses buffered I/O then processing speed can increases.

Demerits Space wastage and inflexibility.

1) Non Contiguous Allocation: In this maximum size of the file does not have to be predicted at the beginning. The file can grows with time as per needs. This reduces wastage of space.

There are two methods of non-contiguous allocation such as

i) Chained Allocation :It is used in non-contiguous allocation. In this each file is a linked list of disk blocks then disk block may be scattered anywhere on the disk. The directory contains a pointer to the first and last block of the file. Create a file in this allocation is easy.

Merits :

- 1) There is no external fragmentation.
- 2) There is no need to declares size of file when it is created.

Demerits:

- 1) Each file requires space for pointer.
- 2) If a pointer is lost then it can not open file.

ii) Indexed Allocation: In this all pointer are brought together into one location is called as indexed block. Each block has its own index block which is an array of disk block address. An index can be a list of pointers. Index allocation support access without suffering from external fragmentation. The index block is normally on the disk block which can be read or written by itself.

Q38) What is segmentation?

Segmentation is similar to that of paging. Pages are physical in nature and are fixed size. Whereas segments are logical divisions of program and they are of variable sizes.

Segments are logical divisions of programs and hence are normally of variable sizes. Segmentation is a memory management scheme which support user's view of memory. Each segment has number and length. Each program in executable form can be considered to be consisting of different segments such as code, data and stack. Each of these can be further divided into new segments.

A program normally contains main program and some subprograms and few predefined and precompiled functions. Each of these belongs to different segments. An application programmer does not necessarily have to declare different segments in the program. If various segments in his programs does not defines explicitly then the compiler does it by its own

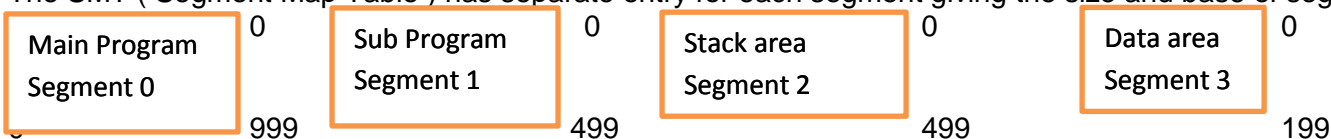
Jobs of compiler :

Followings are the different jobs performed by compiler

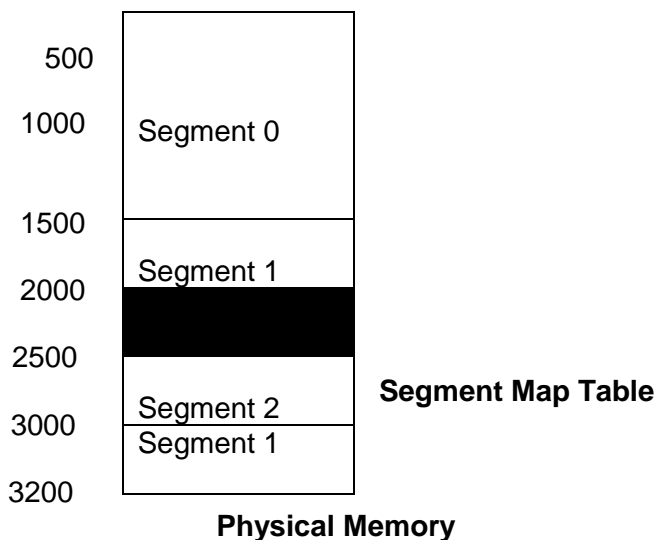
- 1) Compiler recognize different segments in program.
- 2) Number those segments.
- 3) Define segment table.
- 4) Produce an executable image by assigning two dimensional addresses.

Consider following examples as shown in figure there are four segments numbered from 0 To3.

The SMT (Segment Map Table) has separate entry for each segment giving the size and base of segment.



Segment #	Limit	Base
0	1000	500
1	500	1500
2	500	2500
3	200	3000



Q) Explain the concept of virtual memory in brief.

Virtual memory is memory scheme which makes the execution of the processes possible which may completely not in the main memory. Some part of the process may be on disk.

The memory management techniques are simple to implement but the major drawback is that if the physical memory is limited then number of processes it can hold at any time i.e. degree of multiprogramming reduces. For this concept of virtual memory is introduced.

The main advantage of virtual memory is user can execute programs whose size may be greater than the physical memory.

The virtual memory is difficult to implement. It can be implemented by paging, segmentation or combined schemes. Mostly virtual memory systems are implemented using paging.

A program consists of number of logical or virtual pages. To start execution of program, some of the pages are loaded into specific page frames. If a page is not loaded into memory and a location from that page is referenced at that time a page fault is arises. When a page fault arises the operating system loads the referenced page in memory from disk.

Concept of Virtual Memory:

Generally virtual memory is related to following concepts :

1) Locality of reference

The basic principal behind virtual memory is called locality reference. This gives some basis to forecast whether a page likely to be referenced in the near future or not depending on its past behavior in past. Hence it may be removed from memory.

2) Page fault

In many systems when a process is executing with only a few pages in memory and when an instruction is encountered which refers to any instruction or data in some other page which is outside the main memory i.e. on the disk then page fault is occurred. At this stage the operating system must bring the required page into the memory before the execution of that instruction can restart.

3) Working set

At any time a process has a number of pages in the physical memory. Not all of these are actively referred. The set of pages in the physical memory actively referred to at any moment is called as working set. Working set decides the policy of bringing in pages from disk to main memory.

4) Page replacement policy

As the number of processes and the number of pages in the main memory increases all the page frames become occupied. At this time if a new page is to be brought in the operating system has to overwrite some existing page in the memory. The page to be overwritten is selected by page replacement policy. There are number of ways in which the operating system selects the page to overwritten. The operating system designer chooses one amongst many of such policies and writes corresponding algorithm for it.

5) Dirty page / Dirty bit

The page which is modified after it is loaded in main memory from disk is called as dirty page. The operating system maintains one bit for each physical page frame to denote whether a page has become dirty or not. This bit is called as dirty bit.

6) Demand paging:

In demand paging a page is brought in only when demanded. Consider a process is created with no pages in main memory. When the process is dispatched initially then program counter will have been loaded with the address of first instruction. This address obviously belongs to a page outside the main memory. In this way as page fault goes on occurring then operating system brings new referred pages. This is called as demand paging.

Drawbacks:

Demand paging is that a lot of pages which have been used in past but which now are not required in memory unnecessarily.

Q) Define security with respect to an operating system. Explain the different elements of security.

Security is concerned with the ability of the operating system to enforce control over the storage and transportation of data in and between the objects that the operating system supports.

In multiuser operating systems the concept of security and protection are very important because user programs should not interfere with one after another or with the operating system. In general secure systems are those which control through the use of specific security features access to information that only properly authorized individuals or processes operating on their behalf will have access to read, write, create or delete.

Elements of security:

There are three main elements of security such as confidentiality, integrity and availability

1) Confidentiality: Confidentiality ensures that information is not accessed in an unauthorized manner. It is generally related to the read operations.

2) Integrity : Integrity ensures that the information is not amended or deleted by an unauthorized manner. It is generally related to write operations.

3) Availability: It ensures that information is available to the authorized users at right time.

Q) Discuss in brief threats to security in any computing environment.

Sharing and protection are requirements of modern computing environments. But these two are contradictory to each other as more sharing gives rise to possibility of more security threats.

Classification or categories of threats to security:

The major threats to security in any computing environment can be categorized as like follows:

1) Tapping: Unauthorized use of servicing. This is passive threat. **2) Disclosure:** Unauthorized disclosure to information. This is passive threat. **3) Amendment:** Unauthorized alteration or deletion of information. This is active threat. **4) Fabrication:** Unauthorized fabrication of information. This is active threat.

5) Denial : Denial of service to authorized uses. This is active threat.

Q) What are attacks on security? Explain in short the ways in which a system can be attacked.

The security system can be attacked and penetrated in number of ways as like follows:

1) Authentication:

Authentication means verification of access to the system resources. Following are some of the ways in which authentication may take place:

i) By stealing and using somebody incorrect password and then use it.

ii) Use of vendor supplied password which can be used by only system administrator.

iii) Finding password by trial and error method. iv) If a user logs on and then goes off an intruder can use that terminal. v) Writing dummy login programs to fool users.

1) Browsing:

In some system (PC) there exists files with access controls which are very permissive. One can browse through the system file or bootable files to get this information after which unprotected files or databases could be easily accessed. Confidential information could be read or even modified.

2) Trap doors:

Sometimes software engineers leave some secret entry point to modify their programs. These are called as trap doors. They can be misused by others.

3) Electric data capture:

Use of active or passive wire traps or mechanism to pick up the screen radiation and to recognize what is displayed on screen is called as electric data capture.

4) Invalid parameters: Passing invalid parameters may cause serious security violations.

5) Line Trapping : A special terminal is used to tap into a communication line. It causes access to confidential data.

6) Waste recovery : By using some technique deleted files can be recovered password may be recollected.

7) Rough software : Certain programs like worms, viruses attack on system.

8) Lost line: In networking a line may get lost. In such cases operating system can logout a user and allow an access only after reestablishing the identity or user. Some operating system can not do this. In such cases floating process can be accessed by unauthorized user.

9) Improper access control: Some times operating system and administrator may not plan access control properly. This may lead to some users having too many privileges and others very few. It cause disclosure or denial of service.

Q) What are computer worms? Explain its mode of operation.

A computer worm is a complete program. A computer worm can act independently. Generally it do not cause direct harm to the computer system. It just goes on spreading on to network and consumes network resources to a large extent.

Mode of Operation:

A computer worm usually operates on a network. Each node on network maintains a 'mailing list' which contains the names and addresses of the reachable machines on the network. The worm gets accessed to this list and using this it sends a copy of itself to all those addresses.

If the worm is intelligent then after reaching a node it checks whether a copy of itself already exists there or not. If exists it does not creates one more copy.

If the worm is dumb then it just copies itself to all nodes. So if one node's address is at several places in network then it would have several copies of the worm.

Safe guards against worms:

A worm can be prevented by strong security and various check points on the communication system.

Q)What is computer virus? State various types of viruses and the basis on which they are classified.

A computer virus is not a complete program but a part of program. A computer virus can not act independently. It causes direct harm to the computer system. It has been written with clear intention of infecting others. A computer virus corrupts code and data.

Basis of Classification of viruses:

A classification of virus is based on what do it affects or where the virus resides. Thus they are classified as five types as like below:

1) Boot sector virus : Virus can get into system memory if the machine is booted with an infected floppy hard disk.

2) Memory resident virus : It loads upon execution of an infected file. Subsequently whenever a non infected file is executed then virus infects it.

3) File specific virus :Infection occurs when an infected file is executed. The virus then loads its at viral code into memory. In addition to infecting the virus also reboot the system.

4) Command processor virus : Infection is occurred when an infected file is executed. The virus then loads its viral code into memory.

5) General purpose virus :General purpose virus can infect exe i.e executable file.

Q) What are the different methods by which virus can infect other programs?

There are five well known methods by which a virus can infect other programs:

1) Append : In this method the viral code appends itself to the unaffected programs.

2) Replace : In this case the viral code replaces the original executable program completely or partially to carry out some funny actions.

3) Insert : In this case the viral code is inserted in the body of an executable code to carry out some funny or undesirable actions.

4) Delete : In this case the viral code deletes some code from the executable program.

5) Redirect :This is an advanced approach employed by the authors of sophisticated viruses. The normal control flow of a program is changed to execute some other code which could exist as an appended portion of normal program.

Q)How generally virus operated and prevented?

Virus Mode of operation :

Virus works in number of ways. Developer of virus is a bright person who knows the operating system very well. Typically virus are developed under MS-DOS. Viruses are spreads with games, unauthorized softwares and they are distributed to people free of charge.

Virus prevention :

Prevention is best measure. The safest way is to buy official, legal copies of software from reliable sources. Free, unreliable softwares should not be used. Frequent backup and running of monitoring programs also helps prevention

Q) Define : 1) Virus Detection 2) Virus removal 3) Virus prevention

1) Virus Detection : It is a software program that checks for integrity of binary files. The program maintains a checksum on each file. A mismatch in it indicate virus. Some programs reside in memory and continuously monitor memory and I/O operations.

2) Virus Removal : It is a software program that scans disk for pattern of known viruses. For some viruses bit pattern in code can be predicted. On detection it removes them. The data recovery is impossible when virus damages data.

3) Virus Prevention : Prevention is best measure. The safest way is to buy official, legal copies of software from reliable sources. Free, unreliable softwares should not be used. Frequent backup and running of monitoring programs also helps prevention.

Q) Differentiate between worms and viruses.

Computer Worm	Computer Virus
1) A computer worm is a complete program.	1) A computer virus is not a complete program but a part of program.
2) A computer worm can act independently.	2) A computer virus can not act independently.
3) Generally it do not cause direct harm to the computer system.	3) It causes direct harm to the computer system. It has been written with clear intention of infecting others.
4) It just goes on spreading onto network and consumes network resources to a large extent.	4) A computer virus corrupts code and data.

Q) What is GUI? What are advantages of GUI?

The interfaces which replaces cryptic commands by their graphical representation are called as graphical user interface (GUI). The first GUI was developed by XEROX corporation for their Xerox star computer. The Microsoft developed GUI Operating system is called MS-Windows for IBM(International Business Machine) personal computer . Now various GUIs for Unix based systems are also available.

Advantages of GUI :-

1) Easy to use consistent GUI for virtually all programs. 2) User can communicate and exchange data between programs without transferring or copying files. 3) With GUI commands are replaced by graphics. Hence it is not necessary to remember the commands and its meaning. 4) With GUI user can run several programs simultaneously.

Q) What is GUI? Explain in brief features of GUI.

The interfaces which replaces cryptic commands by their graphical representation are called as graphical user interface (GUI). The first GUI was developed by XEROX corporation for their Xerox star computer. The Microsoft developed GUI Operating system is called MS-Windows for IBM(International Business Machine) personal computer . Now various GUIs for Unix based systems are also available.

The windows operating is based on GUI.

Features of GUI :

1) Replacement of command with icons:

- Commands are grouped together in various levels of hierarchy and when the user selects a group further commands in that group are displayed.
- This allows the user to select a command using a cursor and simply checking on it.
- User can select the command and use the application without first having to know about the computer and its working.
- The display of these command sequences takes place graphically.

2) Provide online help:

- GUI based applications provides "help" about various features of the application.
- HELP can assist the user in knowing everything about the application.
- In the windowing environment if user gets confused at any point a help is readily available. This makes GUI based applications more popular and efficient.

Q) What are essential components of GUI? Explain functions of each of components of GUI.

The essential components of GUI are as follows:

- 1) Title bar 2) Control Box 3) Minimize button 4) Maximize button/Restore button
- 1) Close button 6) Menu bar 7) Scroll bars 8) Dialogue box

Functions of GUI component:

- 1) **Title bar :** This is horizontal line on top is the name of the program.
- 2) **Control box :** Without using mouse when we want to change size of window then control box can be used. Control box can be opened by pressing ALT key and then right arrow key.
- 3) **Minimize button:** this button is used to minimize program, without closing it. After clicking the button the program goes to bottom line. After clicking the program it can be resized again.
- 4) **Maximize / Restore button :** The window can be made full screen by clicking this square button. Now this button gets converted into restore button. It is indicated by double square. When we click restore window gets its previous size.
- 5) **Close button:** When we click close button, the window or that application gets closed.
- 6) **Menu bar:** The menu bar appears normally at the top of the window and below the title bar. This consists of different main menus which can be used in program. The main menus consists of different submenus. A menu can be selected by clicking it then pull down menus are appears.
- 7) **Scroll bars:** This allows user to scroll window horizontally and vertically. Scroll bars are used to look at information which is not currently visible in screen by scrolling window horizontally or vertically. A scroll bar consists of a horizontal or vertical scroll area with a slider box and an arrow in a box at each end. Slider box gives hint on size and position of the visible part of object.

8) Dialogue boxes:

Dialogue box is a window used by the application to interact with the user. A dialogue box can also be used to display information or to get user input and also for a combination of these two functions.

Types of dialogue box:

There are two types of dialogue boxes such as

- a) **Model dialogue box :**

A dialogue box where an application can continue only after the user has responded to the dialogue box is called as model dialogue box.

- b) **Modeless dialogue box:**

A dialogue box which allows user to continue without responding to it is called as modeless dialogue box.

Q)What is window? What are the operations that can be performed on a window?

Different applications are shown on screen by icons. The screen is split into different partitions. Each of these partition can be of different sizes. We can run different applications in each partition of the screen and watch the programs of each application in them. Each of these partition is called as window. Each window is independent of the other.

1) Dragging the window:

The position of window on screen can be changed by dragging it. To drag a window select a window by clicking a mouse. Keep left button of mouse pressed and move the mouse pointer to new position on screen. Now release the mouse button. You will find window shifted to new position.

2) Resizing window:

Making change in size of window is called as resizing window. The window can be resized as follows:

- i) Move mouse pointer to right side border of window.
- ii) Observe that the shape of mouse pointer get changed to left-right arrow.
- iii) Now press the left button of mouse and moves the mouse arrow to right side to a new position. Release mouse button. Observe that width of window is increased.

Q) Explain various controls of GUI?

A variety of controls are used in a GUI to enable user to select type of information or to select specific operation to be carried out. That are either buttons or boxes. Some of these control button are as follows:

1) **Push button :**

It is a rectangular button having a label indicating action to be carried out. This button is used to select an action represented by button. This button normally used when one action s to be selected at of many choices.

e.g.

A rectangular button with an orange border and the text "SAVE" inside.

2) **Option button or Radio button :**

It consists of a graphical image which is used to select one object out of several possible objects. The currently selected can be distinguished from the others by highlighting on the graphic image.

Eg.

- One
- Two
- three

3) **Check button :**

A check button consists of a square box and an accompanying text. This is used for selecting one or more choices from a list of options.

e.g

- ☒ One
- ☒ Two
- ☒ Three

4) **List box :**

A list box is a rectangular box with scroll bars. This allows user to select one item from a scrollable list of choices.

5) **Entry box :**

An entry box is a rectangular box which allows user to enter some text. An additional hint about the type of text to be entered is provided near the box.

Q) Explain in brief the following programs of MS-Windows:

i) **Program Manager**

ii) **File Manager**

iii) **Control Panel**

i) **Program Manager :**

- a) The program manager starts executing with MS-Windows.
- b) This provides user interface to start and stop applications.
- c) It is used to organize various applications into different groups.
- d) It also indicates how each group contents are controlled and displayed on the screen.
- e) It is also used to end the MS-Windows session.

ii) **File Manager :**

- a) This helps organize user files and directories.
- b) This is used to traverses through the file system and change drives, to search, copy, move, create, or delete files and directories.
- c) Applications can be started directly from the file manager.

iii) **Control Panel :**

- a) It can be used to choose or change the color schemes in the applications, select and display the background of the screen, select border width and other border characteristics, cursor size and shape etc.
- b) Fonts also managed by control panel.
- c) It also used to configure printers and other ports on the PC.

H.S.C. BOARD QUESTION BANK OBJECTIVE TYPE QUESTIONS

- 1) The element of a record are ----- (Oct 2003)
a) Homogenous b) Similar c) non homogenous d) identical
Ans : c) non homogenous
- 2) The most efficient search algorithm is ----- (Mar 2004)
a) Binary search b) reverse search c) linear search d) pointer search
Ans : a) Binary search
- 3) The number of comparisons are required for bubble sorting of an array of 'n' elements is ----- (Oct 2004)
a) $n(n-1)/2$ b) $n/2$ c) $\log_2 n$ d) $\log_{10} n$
Ans : a) $n(n-1)/2$
- 4) Finding the location of records with a given key value is known as ----- (Mar 2005)
a) Traversing b) Searching c) Sorting d) Inserting
Ans: b) Searching
- 5) Maximum number of nodes of symmetric binary tree with depth x are ----- (Oct 2005)
a) 2^x b) $\log_2 x$ c) x^2 d) $2^x - 1$
Ans d) $2^x - 1$
- 6) Maximum number of nodes in a symmetric binary tree with depth four are ----- (Mar 2006)
a) 4 b) 15 c) 16 d) 5
Ans : b) 15
- 7) Maximum number of nodes of symmetric binary tree with depth of 6 is ----- (Oct 2006)
a) 6 b) 6 c) 63 d) 36
Ans : c) 63
- 8) ----- is operation of rearranging the elements of an array either in increasing or decreasing order.
a) Sorting b) Searching c) DMS d) DBMS
Ans : a) Sorting
- 9) The complete binary tree (T_n) has $n=15$ nodes then its depth (d_n) is ----- (Oct 2007)
a) 2 b) 3 c) 4 d) 5
Ans : c) 4
- 10) Maximum number of nodes of symmetric binary tree with depth of 7 is ----- (Mar 2008)
a) 125 b) 127 c) 128 d) 124
Ans : b) 127
- 11) Elements of array are always ----- (Oct 2008)
a) Homogenous b) Heterogeneous c) Non homogenous d) none a,b,c
Ans: a) Homogenous
- 12) Record contains ----- data. (Mar 2009)
a) Homogenous b) Non homogenous c) Same d) none of a,b,c
Ans : b) Non homogenous
- 13) Maximum number of nodes of symmetric binary tree with depth n are ----- (Oct 2009)
a) 2^n b) $\log_2 n$ c) n^2 d) $2^n - 1$
Ans : d) $2^n - 1$
- 14) Sorted list is essential requirement for ----- process of an array. (Mar 2010)
a) Linear search b) Binary search c) traversing d) insertion
Ans: b) Binary search
- 15) Maximum number of nodes of symmetric binary tree with depth of 6 is ----- (Oct 2010)
a) 31 b) 127 c) 63 d) 64
Ans: c) 63
- 16) Accessing each element only once is called as ----- (Mar 2011)
a) Traversing b) Searching c) Inserting d) Deleting
Ans : a) Traversing

- 17) Finding locations of a given element is called as ----- (Oct 2011)
 a) Traversing b) Inserting c) Searching d) none of a,b,c
 Ans : c) Searching
- 18) Maximum number of nodes of symmetric binary tree with depth n are ----- (Mar 2012)
 a) 2^n b) $2^n - 1$ c) $\log_2 n$ d) n^2
 Ans : b) $2^n - 1$
- 19) The time required to execute the bubble sort algorithm is proportional to ----- (Mar 2013)
 a) N b) $2^n - 1$ c) n^2 d) $n+1 / 2$
 Ans c) n^2

Oct 2003

- 1) Explain with flowcharts the following control structures : (Mar 2009, Mar 2012)
 a) Sequence logic b) Selection logic c) Iteration logic
- 2) With suitable example show a labeled diagram for link between two nodes having the information part and next pointer field. (Mar 2006)
- 3) Explain the advantage of binary search algorithm with a suitable example. State any two disadvantages or limitations of binary search. (Mar 2007)
- 4) With suitable example explain pointer and pointer arrays.

Mar 2004

- 5) What do you mean by term searching? Which are the different types of searching algorithms? Explain the linear searching algorithm. (Oct 2004)
- 6) What are records? Explain how records are represented in memory using arrays.
- 7) Explain what is meant by linked list with suitable example and a properly labeled diagram.

Oct 2004

- 8) What are data structure? What are the different operations that can be performed on data structures.
- 9) Explain three types of control structures used for flow of control. (Oct 2005, Oct 2011)

Mar 2005

- 10) In case of data structure explain the following terms : (Oct 2007, Mar 2009)
 a) Field b) Record c) File
- 11) With suitable example and diagram show a link list with information elements and the link field from start to null pointer. (Oct 2010)
- 12) Explain bubble sort algorithm with a suitable example. (Oct 2005, Oct 2008, Mar 2010, Mar 2011)
- 13) What is a record structure? How it differs from a linear array.
 (Mar 2007, Mar 2008, Oct 2009, Oct 2011)
- 14) What is traversing an array? Give algorithm for traversing a linear array. (Mar 2006, Oct 2012)

Oct 2005

- 15) Explain stack and queue with suitable examples. (Oct 2010, Mar 2013)

Mar 2006

- 16) What is data structure? Write any four different operations that can be performed on data structure.
- 17) Draw the tree diagram which corresponds to the following algebraic expression:

$$E = (2X + Y) / (5A - B)^3$$

Oct 2006

- 18) Explain the following data structure with suitable diagram : (Mar 2012)
 a) Linear array b) Pointer array c) Tree
- 19) What is linked list? How it will be represented in memory.
- 20) What is searching? Explain binary search algorithm with example. (Oct 2011, Oct 2012)
- 21) Explain all six operations performed on a data structure. (Oct 2009, Mar 2010, Mar 2012, Oct 2012)

Mar 2007

- 22) What searching? Write algorithm for linear search.
- 23) What is linked list? Show linked list with a suitable example having six nodes with a properly labeled diagram. (Mar 2008)

Oct 2007

- 24) What is linked list? Show representation of linked list having three nodes in a memory. (Oct 2012)
 25) Write an algorithm to delete one element from a linear array.
 26) Write an algorithm for linear search method with example. (Mar 2009, Oct 2010)

March 2008

- 27) Explain the following data structure with suitable diagram : (Mar 2011)
 a) Linear array b) Linked list c) Tree

Oct 2008

- 28) What is linear array data structure? Explain with suitable diagram and notations the following terms:
 a) Length of an array b) Subscript variable
 29) Show representation of five records with three fields in a memory as a collection of parallel arrays.

Mar 2009

- 30) Write an algorithm for inserting an element to a linear array. (Mar 2011)
 31) What is binary tree? With suitable example show the relationship between total number of nodes and depth of a tree. (Oct 2003, Mar 2006)
 32) What are binary tree? Draw the binary tree structure for the following expressions :

$$E = (a + b) / ((c * d) - e)$$
 (Mar 2004)
 33) What is binary tree? Draw the binary tree structure for the following expression

$$E = (p - q) / ((r * s) + t)$$

 34) What is binary tree? With a suitable example explain the terminology describing family relationship between the elements of a binary tree.
 35) Explain bubble sort algorithm with suitable examples. (Mar 2008, Mar 2012)
 36) Define binary tree. Draw the tree structure for the following algebraic expression:

$$(p * ((q - r) + t) / ((q + r) * s))$$

 37) What is binary tree? Draw the structure for the following expression :

$$E = (a - b) / ((c * d) + e)$$

 38) What is binary tree? Show a tree structure for the expression : $Z = (A + B / E) * (C + D / F)$
 39) What is binary tree? Draw the binary tree structure for the following expression

$$E = ((a + b) * c) / ((a * ((b - c) + a))$$

 40) With suitable example and labeled diagram show linked representation of binary tree in memory. (Mar 2009)
 41) What is binary tree? How it is represented in memory explain with example?
 42) Explain the terms siblings and leaf in case of a tree structure. Draw tree diagram for the expression

$$C = (X / Y)^3 - (5A + B^2)$$

 43) What is tree structure? Draw the diagram for the given expression :

$$A = (P + 2Q)^2 - (R / 3)$$

 44) What is record? Show representation of records in memory by considering suitable example of three records and three fields.
 45) Write the binary search algorithm with suitable example.
 46) What is binary tree? Draw the tree structure for the following expression

$$E = (p + q) / (r * s) - t$$

 47) Draw binary tree structure for the following expression:

$$(2A + B) (5F - D)^3$$

Mar-20	Explain bubble sort algorithm with suitable example.
Mar-20	Explain linked representation of binary tree in memory with suitable example.
Mar-20	List any six data structure operations.
Mar-20	Define Binary Tree. Draw a Tree diagram for following expression: $Y = ((a-b-c) + (a+b-c)3) / 2$
Mar-20	Define Linked List. Draw and explain labeled diagram of linked list with 5 nodes.
Aug-18	What is Traversing? Write an algorithm to traverse the linear array with example.
Aug-18	Explain following terms used in Binary Tree : i) Sibling ii) Successor iii) Leaf
Aug-18	Explain memory representation of linked list with suitable example.
Aug-18	Define Data Structure. Explain following control structure : i) Sequence Logic ii) Selection Logic iii) Iteration Logic
Mar-18	What is Array? Write an algorithm for Traversing Linear Array.
Mar-18	What is Data Structure? Explain Linear Data Structure and Non-Linear Data Structure.
Mar-18	Define following Terms : i) Group Item ii) Elementary Item iii) Entity
Mar-18	What is Binary Tree? With suitable example show the relationship between Total Number of Nodes and Depth of Binary Tree.
Mar-18	What is Record? Explain how records are represented in memory using array?
July-17	Write any three distinguish points between Linear Search and Binary search.
July-17	Define the following terms with respect to data structures : i) Array ii) Stack iii) Queue
July-17	Explain any six data structure operations.
July-17	Draw a binary tree structure for the following expression : $F = (a - b) / ((P*m)+h)$
July-17	What is an algorithm? Write an algorithm to find the roots of a quadratic equation $ax^2 + bx + c = 0$, where $a \neq 0$.
Mar-17	Explain with flowchart the following control structure : i) Sequence Logic ii) Selection Logic iii) Iteration Logic
Mar-17	Explain Bubble Sort Algorithm with suitable example.
Mar-17	Write difference between Linear Search and Binary Search.
Mar-17	How linked list are represented in Memory?
Mar-17	Define the following terms with reference to Tree : i) Root ii) Leaf iii) Sibling iv) Depth
July-16	What is a linear array? With a suitable diagram and notation explain the terms : i) Length of an array ii) Subscripted variable
July-16	With a suitable example and neat diagram show the representation of linked list in memory.
July-16	Explain linear search algorithm with a suitable example.
July-16	With a suitable example show the representation of binary tree in a memory.
July-16	Explain pointers and pointer arrays with a suitable example.
Mar-16	Define : i) File ii) Record iii) Key-field
Mar-16	State algorithm for inserting an element in an Array.
Mar-16	What is a Record? How it is represented in Memory?
Oct-15	List any six operations performed with data structure.
Oct-15	Write an algorithm to insert an element into array.
Oct-15	What is binary tree? With suitable example, give the linked representation of binary tree in a memory.
Oct-15	Write an algorithm for traversing the array.

Oct-15	Draw the binary tree diagram for following expression: $E = ((a + b) - c) * (e / f)$
Mar-15	What is Data Structure? Define Array and Pointer Array in Data Structure.
Mar-15	Write an algorithm to find smallest element in an Array.
Mar-15	Define: i) Tree ii) Binary Tree iii) Extended Binary Tree
Mar-15	Write an algorithm for Binary Search Method. Explain algorithm with suitable example.
Mar-15	With suitable example explain how tree can be represented in Memory?
Mar-15	What is Linked List? How they can be represented in Memory?
Oct-14	What is linked list? Explain with suitable example.
Oct-14	Write an algorithm to sort the element of array using bubble sort method.
Oct-14	Define the following term of data structure: i) Field ii) Record iii) File
Oct-14	Draw the binary tree for following expression: $E = ((a - b) + (b - c)) / e$
Oct-14	Explain the following control structure and their types use in data structure: i) Selection ii) Loop
Mar-14	What is Record? How it differ from an Array?
Mar-14	Distinguish between Linear and Binary Search.
Mar-14	Write an algorithm for Binary Search.
Mar-14	Define binary tree. Draw binary tree diagram for the following expression : $(a^2 + 2ab + 5c) / (a+b)$
Mar-14	What is a Linked List? Explain how it is represented in memory with the help of in example.
Oct-13	Explain the following terms regarding data structure: i) Length of any Array ii) Subscript with suitable diagram of an array A with length 5.
Oct-13	With suitable example and neat diagram show the representation of linked list in memory.
Oct-13	Explain Binary Search algorithm with a suitable example.
Oct-13	What is a Binary Tree? Draw tree diagram for the expression : $B = (3R / 5T) - (R + Q3)$.
Oct-13	Show the representation of records in memory, considering suitable example of three records and three fields.
Mar-13	Explain stack and queue with suitable example.
Mar-13	What is a Linked List? Show a linked list with suitable example having five nodes with a properly labeled diagram.
Mar-13	Explain Binary Search algorithm with a suitable example.
Mar-13	Draw Binary Tree Structure for the following expression: $(2A+B) (5F - D)^3$
Mar-13	With suitable example show representation of records in Memory.
Oct-12	Explain in brief any six data structure operations.
Oct-12	What is traversing an array? Give the algorithm for traversing a linear array.
Oct-12	What is linked list? Show the representation of linked list having three nodes in a memory.
Oct-12	What is Binary Tree? Draw the tree structure for the following expression: $E = (p + q) / (r * s) - t$
Oct-12	Write an algorithm for Binary Search Technique.
Mar-12	Explain the following data structure with suitable diagram: i) Linear Array ii) Linked List iii) Tree
Mar-12	Explain in brief any six data structure operations.
Mar-12	What are pointer array? Explain giving an example.
Mar-12	Write bubble sort algorithm with a suitable example.

Mar-12	Explain with flowchart the following control structures: i) Sequence Logic ii) Selection Logic iii) Iteration Logic
Mar-12	With a suitable example show representation of linked list in Memory.
Oct-11	Explain any three types of control structures used for flow of control.
Oct-11	What is Binary Tree? Draw the tree structure for the following expression: $E = (A - B) / ((C * D) + E)$
Oct-11	Show the representation of records in memory considering suitable example of three records and three fields
Oct-11	Write the Binary Search Algorithm with a suitable example.
Oct-11	What is record? How it differs from a Linear Array?
Mar-11	Write a Bubble Sort Algorithm with suitable example.
Mar-11	In case of data structure, explain the following terms: i) Linear Array ii) Linked List iii) Tree
Mar-11	What is Binary Tree? Explain terminology describing family relationship between the elements of a binary tree with suitable example.
Mar-11	What is record? Show representation of records in memory by considering suitable example.
Mar-11	Write an algorithm for inserting an element to a Linear Array.
Oct-10	Explain stack and queue type data structure with suitable examples.
Oct-10	Explain Linear Search Algorithm with a suitable example.
Oct-10	With suitable example and diagram show a link list with information elements and link field from start to null pointer.
Oct-10	What is a Tree Structure? Draw the diagram for the given expression: $A = (P + 2Q)2 - (R/3)$
Oct-10	What is a Record? How it differs from a linear array.
Mar-10	Explain in brief any six data structure operations.
Mar-10	Show representation of records in memory. Consider suitable example of 4 records and 3 fields.
Mar-10	Explain bubble sort algorithm with a suitable example.
Mar-10	Explain the terms siblings and leaf in case of a tree structure. Draw tree diagram for the expression: $C = (X / Y)3 - (5A + B2)$.
Mar-10	Explain in brief any three advantages of Linked List over Array.

H.S.C. board Probable Marks : 17

1 Mark	Objective	1	1X1	= 01
3 Mark	Question	4	4X3	= 12
4 Mark	Question	1	1X4	= 04

Q1) Define : 1) **Data** 2) **Group item** 3) **Entity** 4) **Elementary item** 5) **Data item**
 6) **Field** 7) **Record** 8) **File**

1) **Data** : Data are simply values or set of values.

2) **Group item** : Data item which are divided into sub item are called as group item.

e.g. date is called group item because it is further divided into day, month, year.

3) **Entity** : An entity that have certain attributes or properties which may be assigned values. These values may be numeric or non-numeric.

e.g. Name Age Div ----→ all are called attributes

 Rahul 21 A -----→ all are called values of each attributes.

4) **Elementary item** :

Data item which are not divided into sub items are called as elementary item.

e.g. pincode is called elementary item because it can not divided into another sub item.

5) **Data item** : Data item refers to a single unit values.

6) **Field** : Field is a single elementary unit of information representing an attribute of entity.

e.g. name, age etc.

7) **Record** : Record is a collection of fields values of a given entity.

8) **File** : File is a collection of records of the entities in a given entity set.

Q2) What is data structure? What are different types of datastructure?

Definition : The logical or mathematical model of a particular organization is called as datastructure.

Types of datastructure :

There are two types of datastructure such as

a) **Linear datastructure** :

In this data elements are stored in a consecutive memory locations or linked representation.

e.g. 1) array is called as linear datastructure because its all elements are stored in consecutive memory location

2) linked list is called as linear datastructure.

b) **Non linear datastructure** :

In this data elements are stored in non linear manner or hierarchical manner.

e.g. Binary tree and stack and queue datastructure are called as non linear datastructure.

Q3) Explain in detail data structure operation?

There are six types of datastructure operations such as follows :

i) **Traversing** :

Accessing or visiting each record or element exactly once. So that it can be processed is called as traversing.

ii) **Inserting** :

Adding or inserting a new record to the existing structure is called as inserting.

iii) **Deleting** :

Removing or deleting a record from the existing structure is called as deleting.

iv) **Searching** :

Finding the location of a record with a given key values or finding the locations of all records which satisfy one or more given conditions is called as searching.

v) **Sorting** :

Arranging records in some logical order such as ascending or descending is called as sorting.

vi) **Merging** :

Merging means combining the records in two different sorted files into single sorted file is called merging.

Q4) What is linear array? How it is represented in memory?

Linear array is a datastructure which contains finite or fixed set of ordered homogenous data elements.

e.g. Let A is linear array having five elements as like below :

100
200
300
400
500

1) The elements array are referenced by an index set are in consecutive memory locations. So that A[0], A[1], A[2], A[3], A[4]. Therefore they are stored in successive memory locations.

2) The elements of an array can be denoted by their subscript notation as like A[0]=100, A[1]=200, A[2]=300, A[3]=400 and A[4]=500.

- **Length of array OR Size of Array :-**

The number 'n' of the elements is called as length or size of an array. In general the size or length of the array can be obtained from the index set by the following formula :

Length or Size of array = $UB - LB + 1$

Where

1) UB – the largest index called as Upper Bound.

2) LB- the smallest index called as Lower Bound.

Representation of Linear Array in memory :-

1) The elements of LA are stored in consecutive memory locations.

2) The computer does not need to keeps track of the address of every element of an array. It just requires the address of first element of an array which is denoted by Base [LA] and called as the base address of linear array LA.

3) Using base address the computer calculates the address of any element of an array using the following formula :

$LOC (LA[K] = Base[LA] + W (K-LB))$

Where, 1) LOC(LA[K]) is an address of k^{th} element of LA.

2) 'W' is number of words per memory location for LA.

3) LB is lower bound i.e. smallest index of LA.

Memory Address

1000

1001

1002

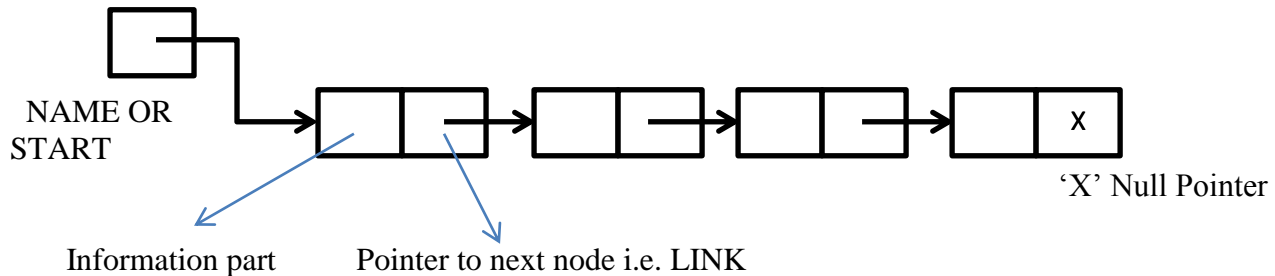
Array elements are stored sequentially

Fig : Memory representation of array

Q5) What is linked list? What are advantages of linked list?

- i) Linked list is a linear collection of data elements called as nodes where linear order is maintained with the help of pointers.
- ii) It is also called as one-way list.
- iii) Each node in the linked list is divided into two parts such as follows :
 - a) First part : The first part of linked list is called as INFO part which contains the information about the elements or actual elements.
 - b) Second part : The second part of linked list is called as LINK part. This is next pointer field i.e. it contains the address of next node in the list.

Linked list Diagram having four nodes :



In above figure left part is called as INFO part which contains information of element. Second part is called as LINK part is right part which is next pointer field it points to next node. The arrows drawn from LINK part of one node to the next node indicate the link.

The address of list is stored in START or Name of list which trace linked list. The LINK part of last node is NULL pointer i.e. it contains nothing.

Advantages of linked list over linear array : The following are advantages of linked list over linear array.

- 1) Linked list can not require consecutive memory locations to store in memory location.
- 2) Linked list easily extended.
- 3) Insertion of any element is easy.
- 4) Deletion of any element is easy.
- 5) Linked list is easily implemented and maintained in computer memory.

Q6) How linked list are represented in memory?

OR With suitable example show the representation of linked list in memory?

To store linked list in memory two arrays are required are called as INFO and LINK. Such that INFO [K] contains information of actual elements in an array while LINK [K] contains next node address. The variable Name or Start of linked list contains address of first node. The pointer field of last node denoted by NULL which indicates the end of list.

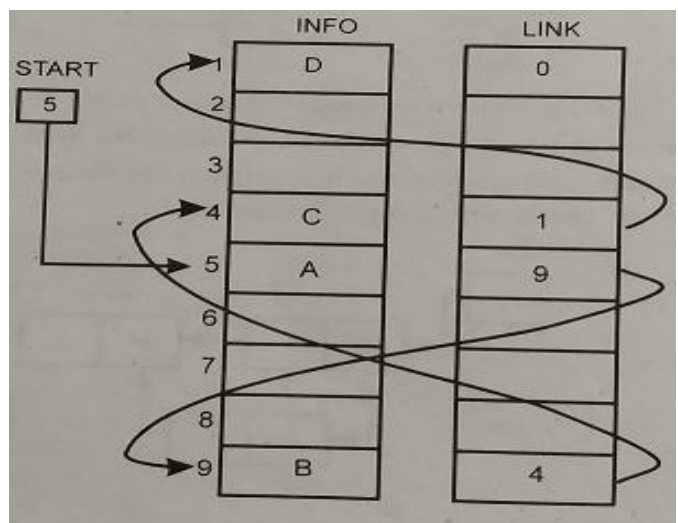
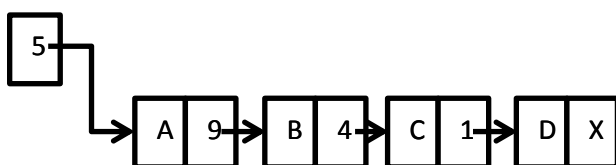


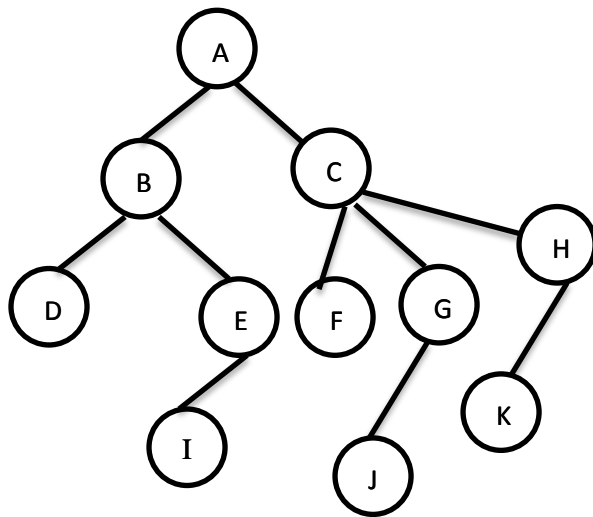
Fig: Memory Representation of Linked list diagram

Q7) What is Binary Tree ? Explain in detail terminology describing family relationship between the elements of binary tree?

The binary tree is non linear, hierarchical data structure which consists of finite set of one or more nodes i.e. collected data item.

***Terminology describing family relationship between the elements of binary tree :->**

The binary tree contains family members such as :



1) Node :-

Each element of binary tree are called as node. In diagram there are 11 nodes. The total number of binary is calculated by formula :
Total number of nodes = $2^n - 1$ Where 'n' is depth or height of binary tree.

e.g. If binary tree have depth 5 then total number of nodes in binary tree is 31 nodes.

2) Root :-

A node which has no parent node is called as root which is a first node. In diagram A is called as root.

3) Leaf :-

This is also called as terminal node. The node which has no child or children is called as leaf or terminal node. This have degree zero. In diagram D, I, F, J, K are leaf or terminal nodes.

4) Child :-

Those node which are subnode or child of parent node is called as child. In diagram A is root or parent node have childrens B and C. Similarly B is parent have childrens D and E. C is parent have childrens F, G and H etc.

5) Sibling:-

Childrens of the same parent are said to be sibling.

e.g. In diagram B and C are called siblings because they are childrens of same parent A. Similarly D and E are called siblings because they are childrens of same parent B. Similarly F, G and H are called siblings because they are childrens of same parent C.

6) Left Successor :- In diagram B is called as left successor of node A.

7) Right Successor :- In diagram C is right successor of node A.

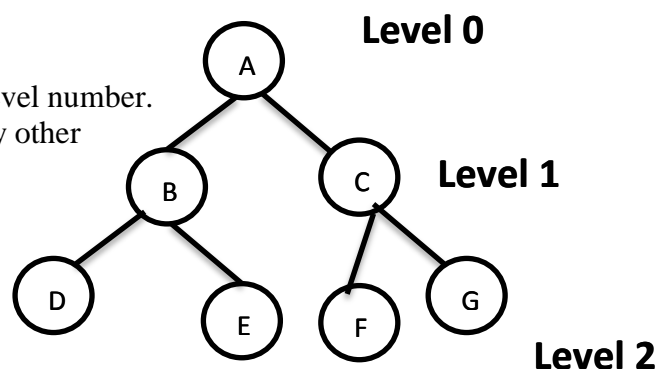
8) Left Subtree :- In diagram B, D, E, I are left subtree.

9) Right Subtree :- In diagram F, G, H, J and K are right subtree.

Q8) Explain following terms of binary tree

1) Level of tree :- Each node have assigned level number.

The root node of tree have level 0 and every other node is assigned to level number which is more than the level number of its parent. Therefore it is the distance from root.



- 2) **Depth or Height of tree :-** Depth or Height of tree is defined as maximum level of any node in a tree + 1.

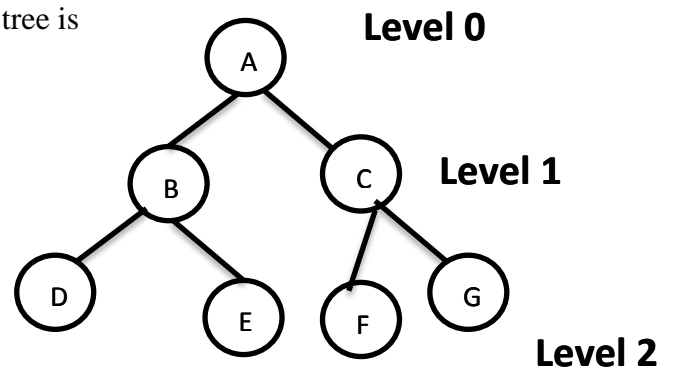
In diagram Maximum level is 2

So that

$$\text{Depth} = \text{Max level} + 1$$

$$= 2 + 1$$

$$= 3$$

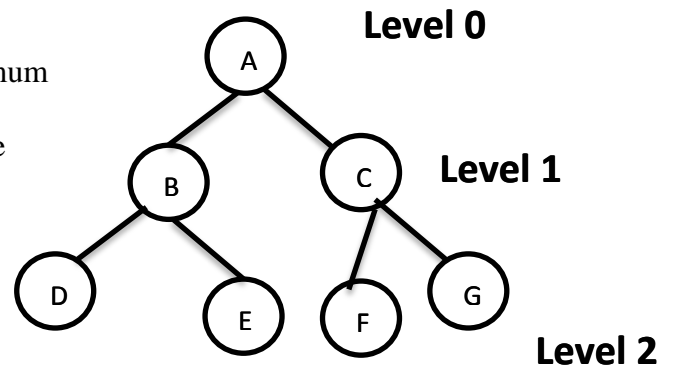


- 3) **Degree of tree :-**

The number of subtrees of node is called as degree of a node. Therefore the maximum degree in tree is degree of that tree.

In diagram degree of each as in table

Node	Degree
A	2
B	2
C	3
D, E, F, G and H	0



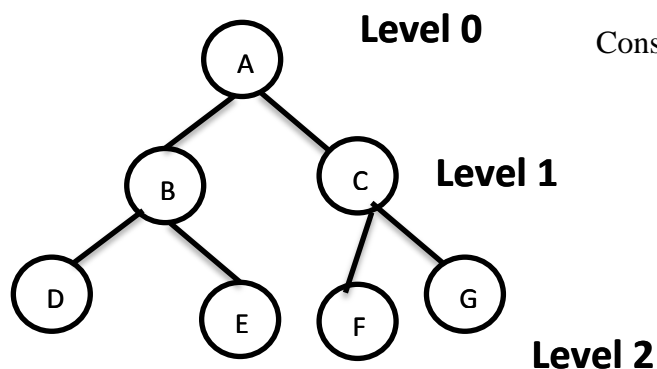
Therefore, in diagram the tree contains maximum degree 3.

So that :

$$\text{Degree of tree} = 3$$

- Q9) **With suitable example show the relationship between total number of nodes and depth of tree?**

The maximum number of nodes of symmetric binary tree with depth 'n' are $2^n - 1$. Where 'n' is depth or height.



Consider the tree as in diagram

In diagram maximum level = 2

$$\text{Depth/Height} = \text{Maximum level} + 1$$

$$= 2 + 1 = 3$$

Total number of nodes in symmetric binary tree is

$$\text{Total nodes} = 2^n - 1 \text{ Where } n \text{ is depth.}$$

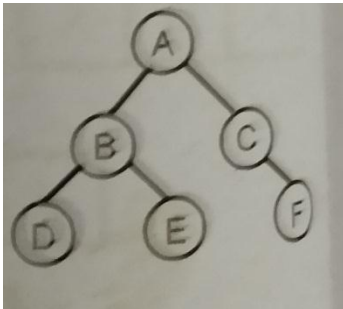
$$= 2^3 - 1 = 8 - 1 = 7$$

Therefore the maximum number of nodes of Symmetric binary tree is calculated by its depth. In diagram it is calculated as 7.

Q10) Define : 1) Complete Binary tree 2) Extended binary tree OR 2 tree ?

1) Complete Binary Tree :-

The tree is said to be complete binary tree if all its level except the last have maximum number of possible nodes.



2) Extended Binary Tree OR 2 tree :-

Level 0

A binary tree is said to be a 2 tree or an extended binary tree if each node has either 0 or 2 childrens.

In such cases the nodes with 2

Level 1

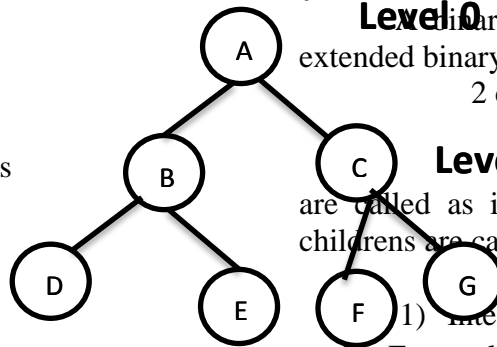
are called as internal node and nodes with zero childrens are called as external node.

In diagram

1) Internal nodes : A and B.

Level 2

2) External nodes : D, E and F and G.



Q11) How binary trees are represented in memory ?

OR With suitable example and labeled diagram show representation of binary tree in memory?

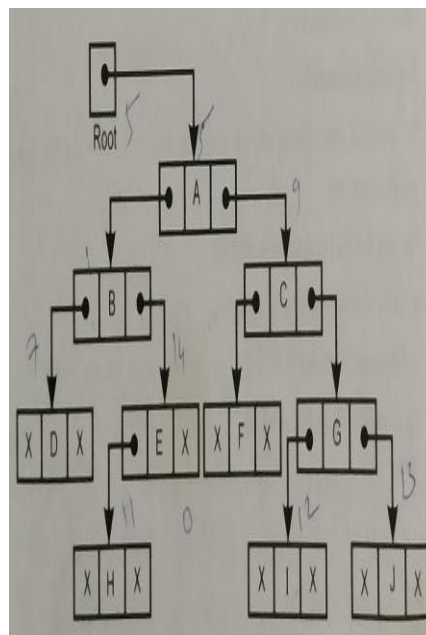
The binary tree is represented in memory by two ways as follows :

1) Linked Representation :-

In linked representation there are three parallel arrays are used to store INFO, LEFT and RIGHT

are three parallel arrays and a pointer variable ROOT.

- INFO [K] contains actual elements.
- LEFT [K] contains address of left child.
- RIGHT [K] contains address of right child.



Computer Science - I
The above tree can be represented in memory as.

	INFO	LEFT	RIGHT
1	B	7	14
2	G	12	13
3		0	
4	E	11	0
5	A	1	9
6		3	
7	D	0	0
8		6	
9	C	14	2
10			
11	H	0	0
12	I	0	0
13	J	0	0
14	F	0	0

Fig a. Linked List Memory Representation diagram

2) Sequential Representation :-

For sequential representation only one linear array is used. This array is generally known as TREE.

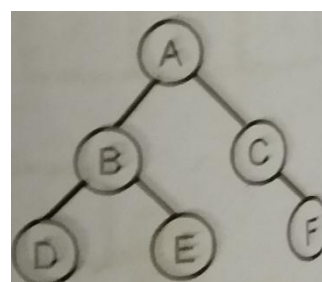


Fig b Sequential Representation Diagram

TREE

1	A
2	B
3	C
4	D
5	E
6	-
7	F
8	-
9	-
10	-

Q12) What is record?

- 1) A record is a collection of relative data item.
- 2) A record is collection of fields i.e. relative data items.
- 3) A collection of records are known as files.
- 4) A record contains non homogenous data items.

Q13) Differentiate between record and linear array?

Record	Linear Array
1) It is collection of non homogenous data items.	1) It is collection of homogenous data items.
2) It is collection of relative data i.e. called as filed.	2) It is collection of linear data elements.
3) Natural ordering is not possible because data item are indexed by fields i.e. attributes.	3) Data item are natural ordered and stores in consecutive or contiguous memory locations and they are sorted in ascending or descending.
4) Referenced by level number.	4) Reference by index set.

Example of record

Suppose a college keeps a record of each students which contains following data item.

ITEM	SUBITEM
1) Name	First, Middle and Last
2) Birthdate	Date, Month and Year
3) Address	City and pincode
4) Phone	Landline and Mobile

The structure of above record is as described as like below :

1. Student
2. Name
 3. First
 3. Middle
 3. Last
2. Birthdate
 3. Day
 3. Month
 3. Year
2. Address
 3. City
 3. Pincode
2. Phone
 3. Landline
 3. Mobile

In this structure the number to left of item name i.e. identifying represents a level number. The subitem followed each group item and the level of subitem is 1 more than level of group item.

**** How to access item in record :-**

Suppose we want to access last name of student then it is accessed by using dot operator as like below

Student.Name.Last

Q14) How records are represented in memory using array?

To represent record in memory array are used. One separate linear array is used for each elementary item of records. The arrays are parallely used.

Therefore the records are stored in memory using parallel linear arrays. All parallel arrays is for subscript k^{th} the elements as below:

e.g. Name [k] , Address [k] , Phone [k] .

Consider suitable example of three records and three fields as like follows :

Name	Address	Phone
RAM	LATUR	9890445566
SHAM	OSMANABAD	8448676662
MOHAN	NANDED	7567456778

Therefore to store above three records and three fields in memory three parallel arrays are required as like follows

Name [k]	Address [k]	Phone [k]
RAM	LATUR	9890445566
SHAM	OSMANABAD	8448676662
MOHAN	NANDED	7567456778

Q15) What is algorithm?

An algorithm is a finite or fixed step by step list of well-defined instructions for solving a particular problem.

*** TWO parts of algorithm :-**

Any algorithm have two main parts such as

a) First part :-

The first part of an algorithm tells the purpose of algorithm. In this part we defines variables in algorithm and lists the input data.

b) Second part :-

The second part of an algorithm consists of steps in algorithm that are executed one after another. Generally beginning with step 1 unless stated otherwise. The control can be transferred to step n by the statement goto step n.

The algorithm is completed when the statement EXIT or STOP is encountered or initiated.

Example : An algorithm to find largest element in array :

Largest(DATA , N , MAX)

Here, DATA is a linear array with N elements. This algorithm finds the largest element by MAX of DATA.

Step 1 : [Initialize counter]
Set $k := 1$ and $MAX := DATA[1]$

Step 2 : [Compare and Update]
if $MAX < DATA [k+1]$
then $MAX := DATA [k+1]$
[End of if structure]

Step 3 : [Increment counter]
Set $k := k + 1$

Step 4 : [Test counter]
if $k < N$ then goto step 2
[End of if structure]

Step 5 : Write : MAX

Step 6 : EXIT

Q16) Explain with flowcharts the following control structures :

1) Sequence Logic 2) Selection Logic 3) Iteration Logic

1) Sequence Logic OR Sequential flow :-

In sequence logic modules i.e. functions are executed sequentially one after another. The sequence may be present explicitly by means of numbered step or by the order in which modules are written.

In short in sequential logic or sequential flow modules of an algorithm are executed one after another.

2) Selection Logic OR Conditional flow :-

Selection logic uses number of conditions which causes Selection of one out of several alternative modules. The structure which implement this type of logic is known as selection logic OR conditional flow structure.

Different types of selection logic or conditional flow :

There are three types of conditional flow or selection logic

i) Single Alternative :-

This has following form or syntax

if condition

then [module A]

[end of if structure]

If condition is satisfied i.e TRUE then module A which consists of statements is executed. Otherwise module A is skipped and next module of algorithm is executed.

ii) Double Alternative

This has following form or syntax

if condition

then [module A]

Else :

[module B]

[End of if structure]

If condition is satisfied i.e TRUE then module A which consists of true statements is executed. Otherwise module Bis executed which contains false statements.

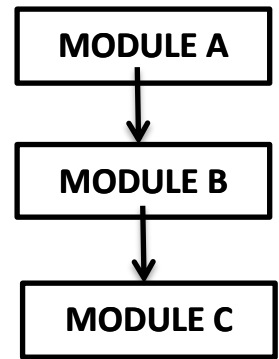


Fig flowchart of sequence logic

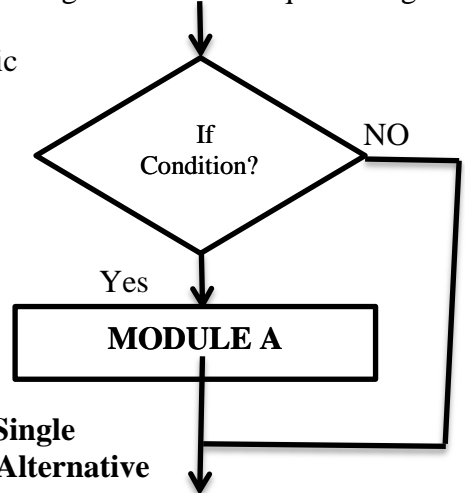


Fig Single Alternative

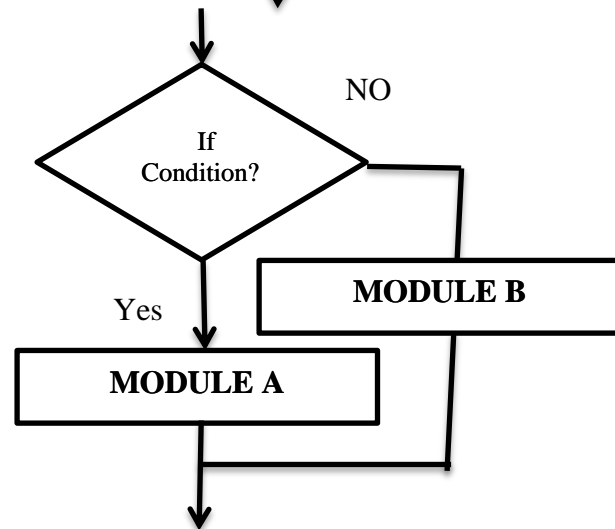


Fig Double Alternative

iii) Multiple Alternative :-

This structure has the following form
if condition (1) then [module A1]
else if condition(2) then [module A2]

else if condition(n) then [module An]
else [module B]
[end of if structure]

The logic of this structure allows only one module to be executed. The module following the condition which is satisfied then the given condition will be executed. If no condition is satisfied then the module which follows last else statement will be executed.

3) Iteration Logic OR Repetitive flow :-

These are two types such as

i) Repeat for loop :-

The repeat for loop has the following general form or syntax

Repeat for K = R To S by T :
[module]
[end of for loop]

Here, K is called as index variable, R is initial value of K

S is final value of K. T is increment or decrement of value K.

In this loop is begins with index value which is initiated with initial value then given condition is satisfied then body of loop is executed then value of T is increment or decrement whatever it is specified in loop then initial value is changed thus loop is continues number of times until condition is satisfied. When condition is false then loop is break.

ii) Repeat while loop :-

The repeat while loop has the following general form or syntax

Repeat while condition :
[module]
[end of for loop]

Here body of loop i.e. module is executed repeatedly until given condition is satisfied.

Q17) What is traversing an array? Write the algorithm for traversing a linear array ?

Traversing an array means accessing with each element of array only at once so that it can be processed.

Algorithm of traversing a linear array

Here LA is a linear array with lower bound LB and upper bound UB. Following algorithm apply operation PROCESS to each element of LA.

Step 1 : [initialize counter]

Set $K := LB$

Step 2 : Repeat step 3 and step 4

While $K \leq UB$

Step 3 ; [visit element]

Apply PROCESS to $LA[K]$

Step 4 : [Increment counter]

Set $K := K + 1$

[End of step 2 loop]

Step 5 : EXIT

OR

This algorithm traverses a linear array LA with lower bound LAB and upper bound UB.

Step 1 : Repeat FOR $K = LB$ TO UB

Apply PROCESS to $LA[K]$

[End of loop]

Step 2 : EXIT

Q18) What is inserting? Write an algorithm for inserting an element to a linear array.

Inserting refers to the operation of adding an element to the existing elements of array. The element can be easily inserted at the end of array. But for insertion in the middle of array it is required to move the elements of array one byte forward.

The following algorithm inserts a data element ITEM into the k^{th} position in an array LA with N elements.

***Algorithm for inserting an element to linear array :-**

INSERT (LA , N , K, ITEM)

Here, LA is a linear array with N elements and K is a positive integer such that $K \leq N$.

This algorithm inserts an element ITEM at K^{th} position in LA.

Step 1 : [initialize counter]

Set $J := N$

Step 2 : Repeat step 3 and step 4

While $J \geq K$

Step 3 : [Move J^{th} element forward]

Set $LA [J + 1] := LA [J]$

Step 4 : [Decrement counter]

Set $J := J - 1$

[End of step 2 loop]

Step 5 : [insert the element]

Set $LA [K] := ITEM$

Step 6 : [Reset N]

Set $N := N + 1$

Step 7 : Exit

Q19) What is deleting? Write an algorithm for deletion of an element from an array?

Deleting means removing an element from the existing elements of an array. Deletion at the end of an array is easier. But if to delete an element from middle of array then to move the elements of array one location upward.

***Algorithm for deletion of an element from linear array:-**

DELETE (LA, N, K, ITEM)

Here, LA is a linear array with N elements and K is a positive integer such that $K \leq N$.

This algorithm deletes K^{th} element from LA and assigns it to variable ITEM.

Step 1 : Set ITEM := LA [K]

Step 2 : Repeat for J = K to N – 1

[Moves (J+1) element backward]

Set LA [J] := LA [J + 1]

[End of loop]

Step 3 : [Reset number N of elements in LA]

Set N := N – 1

Step 4 : Exit

Q20) Explain bubble sort algorithm with suitable example.

Bubble Sort (DATA , N)

Here, DATA is linear array with N elements. This algorithm sorts elements of DATA in ascending order.

Step 1 : Repeat step 2 and step 3

for K := 1 To N – 1

Step 2 : Set Ptr := 1

Step 3 : repeat while Ptr ≤ N – K

a) If DATA [Ptr] > DATA [Ptr +1]

then interchange such as

DATA [Ptr] and DATA [Ptr + 1]

[End of if structure]

b) [increment pointer]

Set Ptr := Ptr + 1

[End of inner loop]

[End of outer loop]

Step 4 : Exit

In above DATA is an array of N elements. Sorting these elements in ascending order means arranging the elements such that :

DATA [1] ≤ DATA [2] ≤ ... ≤ DATA [N]

In bubble sort compare DATA[1] with DATA[2] and exchange then if DATA[1] > DATA[2]. Next DATA[2] is compare with DATA[3]. They are exchanged if necessary. This process is repeated until DATA[N-1] is compared with DATA[N].

One makes N–1 comparisons this is called as **pass**. After the first pass the largest element is sink to the last position. During the next pass compare elements upto the last but one and second largest element moves to the (N-1)st position.

After N–1 passes all elements are sorted in ascending order.

Example

For Descending order sorting instead > sign < sign is used.

Q21) What do you mean by term searching? Which are the different types of searching algorithms? Explain the linear searching algorithm.

Def : Searching means to find out particular element from a given list of elements or checks whether required element is present or not in an array.

There are two types of searching as follows:

- 1) Linear Search
- 2) Binary Search

***Algorithm for linear search technique :-**

LINEAR (DATA , N, ITEM , LOC)

Here, DATA is a linear array with N elements and ITEM is given element. This algorithm finds or searches the location of LOC of ITEM in DATA or sets LOC = 0 i.e. NULL if search is unsuccessful.

Step 1 : [Insert ITEM at the end of DATA]

Set DATA [N + 1] := ITEM

Step 2 : [Initialize counter]

Set LOC := 1

Step 3 : [Search for item]

Repeat while DATA [LOC] \neq ITEM

Set LOC := LOC + 1

[End of loop]

Step 4 : if LOC = N + 1

then Set LOC := 0

[Search is unsuccessful]

Step 5 : Exit

Example : Given DATA array with following 5 elements

11 22 33 44 55

Suppose search element is 33

ITEM = 33

Step1 : Set DATA [6] = 33

New list is 11 22 33 44 55 33

Step 2 LOC = 1

Step 3 Since DATA [1] = 11 \neq 33 \therefore LOC =2

Since DATA [2] = 22 \neq 33 \therefore LOC =3

Since DATA [3] = 33 = 33 = ITEM

Step 4 : Hence ITEM =33 is found at position
LOC=3

Q22) Write an algorithm for binary search technique with example.

Binary search is used to search an element from sorted array is either ascending OR descending order.

Binary (DATA , LB , UB , ITEM , LOC)

Here, DATA is a sorted array is either ascending order list or descending order list with lower bound LB and upper bound UB. Item is given element is search element. BEG denotes beginning element in array. MID denotes middle element of array. END denotes end element location of array DATA.

This algorithm finds or searches the location LOC of ITEM in array DATA or sets LOC = 0 i.e. NULL if search is unsuccessful.

Step 1 : [initialize variables]

Set BEG := LB, END := UB and

MID := INT ((BEG + END) / 2)

Step 2 : Repeat step 3 and step 4

While BEG := END

AND DATA [MID \neq ITEM

Step 3 : if ITEM < DATA [MID]

then Set END := MID – 1

else

Set BEG := MID + 1

[End of if structure]

Step 4 : Set MID := INT((BEG +END) /2)

[End of step 2 loop]

Step 5 : if DATA[MID] = ITEM

then Set LOC := MID

else

LOC := 0 i.e. NULL

(if search is unsuccessful)

[End of if structure]

Step 6 : Exit

Example Given DATA be the following sorted 13 elements in array

11 22 30 33 40 44 55 60 66 77 80 88 99

Suppose search element = ITEM = 40

Step 1 : Initially BEG = 1 and END 13

Hence MID = INT ((1+13)/2) = 7

And so DATA[MID]=DATA[7]=55

Step2: Since 40 < 55 END has its value changed by

END = MID =INT((1+6)/2) = 3

And so DATA[MID]=DATA[3]=30

Step 3: Since 40>30 BEG has its value changed by

BEG=MID+1=3+1=4

Hence MID=INT((4+6)/2) = 5

And so DATA[MID]=DATA[5]=40

\therefore Found ITEM in location LOC = MID = 5

Q23) What are advantages and disadvantages or limitations of binary search.

*** Advantages of Binary Search :-**

The following are advantages of binary search:

- 1) Binary search algorithm is efficient as the search scope gets reduced to half the size of the array with each iteration or loop.
- 2) The number of comparisons required are approximately equal to $\log_2 n$ which are less than linear search.

***Disadvantages of Binary Search:-**

The following are limitations of binary search over linear search:

- 1) The given list must be sorted either ascending or descending order.
- 2) The access of list must be random means the middle element can be accessed.
- 3) At each iteration middle entry calculation is required.

Q24) Write a difference between Linear search and Binary Search?

Linear Search	Binary Search
1) Linear Search performs on unsorted list of elements as well as sorted list elements.	1) For binary search the elements in array are sorted in alphabetically or numerically in sorted manner.
2) It compares the desired element with all elements in an array until the match is found.	2) It compare the value of midpoint with desired value. If the value is greater than midpoint value then first half is checked otherwise second half is checked until search is successful or interval is empty.
3) Insertion of an element in an array can be performed very efficiently when array is not ordered.	3) An insertion of a new element requires that many elements be physically moved to preserved order.
4) For large size of array time required for this search is very large.	4) For large size of array comparatively time required is less.
5) Time complexity is as follows : Worst case: N comparisons Best case : one comparison	5) Time complexity is as follows: Worst case: $\log_2 n$ comparisons Best case: one comparison

Q25) What are pointer arrays?

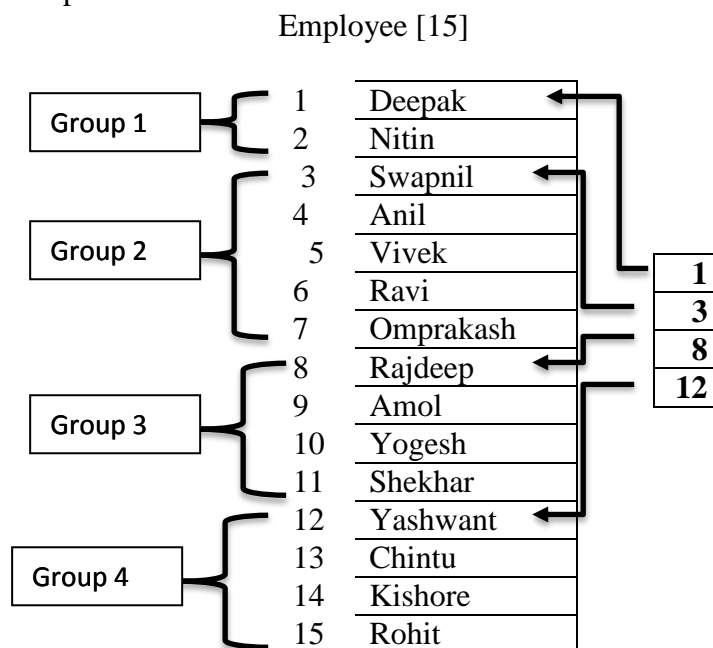
An array is called as pointer array if each element of that array is a pointer. The variable is called as pointer variable if it points to another variable i.e. it contains the memory address of another variable.

Consider an organization which divides its employee list into four groups depending on certain conditions. The following figure shows the list of four groups. There are fifteen employees and groups contains 2,5,4 and 4 employees respectively as follows:

Group 1	Group 2	Group 3	Group 4
Deepak	Swapnil	Rajdeep	Yashwant
Nitin	Anil	Amol	Chintu
-----	Vivek	Yogesh	Kishore
-----	Ravi	Shekhar	Rohit
-----	Omprakash	-----	-----

If above grouped are to be represented memory then the most efficient way is to use two arrays. The first array is Employee which contains list of employees in all four groups sequentially or consecutively. The second array is **Group** which is a **pointer array** which contains the starting address of each group in the Employee array respectively.

The following figure shows as pointer representation :



In above each element of Group array is a pointer, which holds the starting address of different groups. Hence it is called as pointer array.

Q26) Explain stack and queue with suitable example.

OR Explain LIFO and FIFO systems with suitable example.

1) LIFO system OR STACK :-

LIFO system is a last in first out system. In this type of system the elements which is inserted at last and will be deleted first.

Stack is example of LIFO system. It is linear system in which insertion and deletion takes place only at one end i.e. top of the list.

The insertion operation referred as **PUSH** and deletion operation as **POP**.

Example : Consider a stack of dishes. If we want to add a new dish to this stack then it is added at the top of stack also deletion takes place from the top.

2) FIFO system OR QUEUE :-

A FIFO system is first in first out system. In this type of system the element which is inserted first in the list will also be deleted first.

Queue is example of FIFO system. A queue is a linear list in which insertion takes place only at one end of the list known as **REAR** of the list and deletion takes place at the other end is called as **FRONT** of the list.

Example : A queue for tickets in a cinema hall.

H.S.C. BOARD QUESTION BANK OBJECTIVE TYPE QUESTIONS

- 1) The ability to take more than one form is called as ----- in object oriented programming. (Oct 2003, Mar 2011)
a) Inheritance b) encapsulation c) polymorphism d) data abstraction
Ans : c) polymorphism
- 2) Which of the following is not a feature of object oriented programming? (Mar 2004)
a) Follows bottom up approach in program design
b) Objects may communicate with each other through functions
c) Follows top down approach in program design
d) Programs are divided into what are known as objects
Ans : c) Follows top down approach in program design
- 3) Programming in C++ using classes is called ----- programming. (Oct 2004)
a) Procedure oriented b) Event driven c) Object oriented d) database
Ans : c) Object oriented
- 4) ----- is not a derived data type in C++. (Mar 2005)
a) Class b) Arrays c) Functions d) Pointer
Ans : a) Class
- 5) ----- is not the feature of OOPs. (Oct 2005)
a) Polymorphism b) Inheritance c) Data abstraction d) Top down approach
Ans : d) Top down approach
- 6) A derived class with several base classes is called as ----- inheritance. (Mar 2006)
a) Single b) Multiple c) Multilevel d) Hierarchical
Ans : b) Multiple
- 7) To read data from a file the file should be opened in ----- mode. (Oct 2006)
a) Input b) Output c) Append d) none of a,b,c
Ans : a) Input
- 8) Which of the following allows to access the private data of other class is ----- (Mar 2007)
a) Inline function b) Friend function c) Main function d) all a,b,c
Ans : b) Friend function
- 9) While accessing the number in a float array using pointer the pointers value every time is increased by -
----- (Oct 2007)
a) 2 b) 4 c) 8 d) 16
Ans : b) 4
- 10) Following operator can not be overloaded. (Mar 2008)
a) ++ b) :: c) - d) *
Ans : b) ::
- 11) ----- type of member function of class never takes any argument. (Oct 2008)
a) Constructor b) Destructor c) Operator d) none of a,b,c
Ans : b) Destructor
- 12) A derived class several base classes is ----- inheritance. (Mar 2009)
a) Single b) Multiple c) Multilevel d) Hierarchical
Ans : b) Multiple
- 13) Char type data takes ----- bytes in memory in C++. (Oct 2009)
a) 1 b) 2 c) 4 d) 8
Ans : a) 1
- 14) ----- is not a feature of OOP. (Oct 2009)
a) Data abstraction b) Inheritance c) top down approach d) polymorphism
Ans : c) top down approach
- 15) For a=23 and b=3 the value of c after execution of the statement $c=(a/b) * (a\%b)$ will be -----
a) 14 b) 49 c) 21 d) 69 (Mar 2010)
Ans : a) 14
- 16) ----- is not a derived data type in C++. (Oct 2010)
a) Array b) Function c) Pointer d) class
Ans : d) Class
- 17) ----- is not a visibility label. (Oct 2011)
a) Public b) Private c) Protected d) Reserved
Ans : d) Reserved

18) Out of the following C++ operators ----- operator can be overloaded. (Mar 2012)

- a) sizeof b) :: c) ? : d) *

Ans : d) *

19) A pointer is a variable that holds ----- of another variable. (Oct 2012)

- a) Value b) Memory Address c) Datatype d) none of a,b,c

Ans b) Memory Address

20) Object oriented programming follows ----- approach in program design. (Mar 2013)

- a) Top down b) non hierarchical c) random d) bottomup

Ans d) Bottom up

Oct 2003

1) State any six features of object oriented programming. (Mar 2007, Oct 2011)

2) State any three important features of C-string in C++.

3) What is the function of each of the following file stream classes : (Mar 2008)

- a) Ifstream b) ofstream c) filebuf

4) Write any three characteristics of friend function?

Mar 2004

5) Explain use of scope resolution operator and memory management operators in C++ with examples.

6) Explain how memory address of a variable can be accessed in C++. (Oct 2004, Mar 2007, Oct 2012)

7) Explain following OOPS concepts with an example of each :

- a) Inheritance b) Polymorphism c) Data Abstraction

8) What are pointers in C++? Explain the use of pointer variables for function definitions using call by value and call by reference.(Mar 2007, Mar 2008)

9) What is classes in C++? How are member functions defined inside and outside the class. Explain with example.

Oct 2004

10) Describe briefly the features of I/O system supported by C++ with a suitable example.

11) Explain following OOPS concept with an example of each :

- a) Polymorphism b) Data abstraction c) Object

12) What is a class in C++? How are member functions defined inside and outside the class. Explain with examples of each type. (Mar 2008)

13) What are constructors? What are the syntax rules for writing constructors?

Mar 2005

14) State any six important features of object oriented programming.(Oct 2009)

15) Explain in short the three special characteristics of a static data member in a class.
(Oct 2007, Mar 2011)

16) What is constructor in C++? With suitable example explain how a constructor is declared and defined.
(Oct 2010)

17) Explain three types of data conversion in C++ with a suitable example.(Mar 2010)

Oct 2005

18) Explain the difference in between procedural programming approach and object oriented programming.

19) What is constructor and destructor? Give one example of each. (Mar 2006, Mar 2007, Mar 2012)

20) What is class? Give the syntax (form) of class declaration in C++.
(Mar 2008, Mar 2010, Mar 2011, Oct 2011, Mar 2013)

Mar 2006

21) Enlist the basic datatypes used in C++ with size of data in terms of bytes for each.(Oct 2009)

22) Explain with a diagram the file input and file output streams in C++. (Mar 2009)

23) Explain following terms related to object oriented programming with example: (Mar 2008)

- a) Object b) Classes c) Polymorphism

Oct 2006

24) Write a short note on basic data types with their byte sizes.

25) Explain call by value and call by reference with one example of each.

26) Explain the concept of function overloading with one example. (Mar 2008)

27) Explain the following concepts related to object oriented programming :

- a) Objects b) Classes c) Encapsulation

Oct 2007

28) Explain the general concepts of the following in C++ :

- a) Objects b) Classes c) Inheritance

29) Write declaration for each of the following in C++ :

- a) An array of 8 floats b) A pointer to an array of 8 doubles
c) Function that return a pointer to float

30) What is constructor and destructor? State the difference between them.(Mar 2011)

Oct 2008

- 31) What do you mean by object based programming languages and object oriented programming languages? State the relationship between these languages. (Mar 2010)
- 32) Write a declaration for each of the following:
- a) A pointer to an array of 8 floats
 - b) A function that returns a float
 - c) An array of 8 pointers to float
- 33) List any six characteristics of constructor in C++.
- 34) State any three special characteristics of friend function.

Mar 2009

- 35) Define the term 'object oriented programming'. Enlist any four features of object oriented programming.
- 36) What is destructor? What are rules for writing the destructor functions.
- 37) Explain call by value and call by reference with one example of each. (Oct 2011)

Oct 2009.

- 38) What is a constructor? Explain copy constructor with example.

Oct 2003

- 39) Explain operator overloading with illustration. Write the advantages of operator overloading.
- 40) What is polymorphism? How is the same achieved at : i) Compile time ii) Run time (Mar 2006, Mar 2010, Mar 2013)

Mar 2004

- 41) What are classes in C++ for file stream operation? How do you open and close files in C++. Explain any four file modes. (Mar 2012)
- 42) What is operator overloading? Explain with suitable example. Why is it necessary to overload an operator. (Oct 2004)

Oct 2004

- 43) What is inheritance? Explain any three types (forms) of inheritance with suitable diagrams. (Oct 2008, Oct 2010)

Mar 2005

- 44) Explain in brief with suitable diagrams the following terms used in C++ : (Mar 2009)
- a) Multiple inheritance
 - b) Hierarchical inheritance
 - c) Multilevel inheritance
 - d) Hybrid inheritance
- 45) State the details of the following file stream classes:
- a) ifstream
 - b) ofstream

Oct 2005

- 46) State any eight basic rules for virtual functions that satisfy the compiler requirements. (Mar 2007, Oct 2007)
- 47) Write down characteristics of virtual functions.

Mar 2006

- 48) What is operator function? Describe the syntax of an operator function. Explain the difference between operator function as a member function and as a friend function. (Mar 2007, Mar 2011)

Oct 2006

- 49) What is inheritance? Write different forms of inheritance with figure.
- 50) What is polymorphism? Explain compile time and runtime polymorphism. (Mar 2009)

Mar 2007

- 51) State any eight rules for overloading the operators in C++. (Oct 2010, Mar 2013)

Mar 2008

- 52) Explain the concept of function overloading with example.

Oct 2008

- 53) What is an operator overloading? Explain three steps involved in the process of overloading an operator. (Mar 2010, Oct 2011)

Oct 2009

- 54) Explain runtime and compile time polymorphism. Give any one type of example of each.
- 55) What is class? Explain inside and outside class definition with example.

Oct 2003

- 1) Write a C++ program that right justifies text. It should read and echo sequence of left justified lines and print them in right justified format. (Oct 2006, Mar 2007)
- 2) Implement a circle class. Each of object of this class will represents a circle, accepting its radius value as float. Include an area() function which will calculate the area of circle.(Oct 2005, Mar 2008)
- 3) Write a function that uses pointer to search for the address of a given integer of a given array. If the given integer is found then function returns its address otherwise it returns NULL.
- 4) Write a C++ program to accept a number and test whether it is prime or not.
(Mar 2007, Oct 2008, Mar 2011, Mar 2012)

Mar 2004

- 1) Write a c++ to replace every space in an inputted string is less than 80 characters with a hyphen i.e -.
(Mar 2007, Oct 2010, Oct 2012)
- 2) Write a C++ program to find factorial of a natural number inputted during program execution.
(Mar 2008, Oct 2009, Oct 2012)
- 3) Write a C++ program to display a series of 15 terms of the Fibonacci series.
(Mar 2007, Mar 2009, Oct 2011)
- 4) Write the output of the following C++ program (Oct 2007)

```
#include<iostream.h>
long comb(int n, int k) ;
int main( )
{
    const int m = 5;
    for(int i=0; i<m; i++)
    {
        for(int j=1; j<m;j++)
            cout<<setw(2)<<" ";
        for(int j=0; j<=i; j++)
            cout<<setw(4)<<comb(i,j);
        cout<<endl;
    }
}
long comb(int n, int k)
{
    if(n<0 || k<0 || k>n)
        return 0 ;
    long c=1;
    for(int i=1; i<=k; i++, n --)
        c=c*n / i ;
    return c ;
}
```

Oct 2004

- 1) Write a C++ program to exchange the contents of two variables using call by reference.
- 2) Write a C++ program to find factorial of a natural number is inputted during program execution.
- 3) Write a program in C++ to read a set of numbers from keyboard and find out the largest number in the given array.
- 4) Write a C++ program to find greatest common divisor (GCD) of two natural numbers. (Mar 2006, Mar 2011)

March 2005

- 1) Write a C++ program that inputs and stores 10 numbers in an array and prints the sum and average of the array elements.
- 2) Write a function that uses pointer to search for the address of a given integer of a given array. If the given integer is found then function returns its address otherwise it returns NULL.
- 3) Write a program that right justifies the text. It should read and echo a sequence of left justified lines and print them in right justified format. (Mar 2006)
- 4) Write a C++ program to find the smallest of four given integers using min() function that returns the smallest of four given integers int min (int, int, int, int) ;

Oct 2005

- 1) Write a C++ program to display content of an array using pointer.
- 2) Write a C++ program to accept the string from the user and reverse a string.
- 3) Write a C++ program to find the greatest common divisor (GCD) of two numbers. Define a method find to accept the values and calculate greatest common divisor (GCD) of two numbers and print the GCD value.
(Mar 2008)

Mar 2006

- 1) Implement a class temperature to covert degree Fahrenheit value to degree Celsius value.
[hint : $C/5 = F - 32/9$] where C is temperature in degree Celsius and F is temperature in Fahrenheit degree.
- 2) Write the following power () function in C++ that returns x raised to the power n where n can be any integer
double power (double x, int p) ;

Use the algorithm that would compute x^{20} by multiplying x by 20 times.

Oct 2006

- 1) Write a C++ to read a set of numbers form the keyboard and to find out largest number in the given array. The number are stored in a random order.
- 2) Write a function that uses pointer to copy an array of double..
- 3) Write a function that has passed an array of n pointers to floats and returns a pointer to the maximum of a floats.

Oct 2007

- 1) Write a C++ program with computetriangle() that returns the area a and perimeter p of a triangle with given side lengths x, y and z. Use following function prototype in program:
Void computetriangle (float &a, float & p, float x, float y, float z) ;
- 2) Write a C++ program with computecircle() function that returns the area a and circumference of circle c with given radius r.
- 3) Write a C++ program to print the input string in a reverse order using function which first locates the end of string then it swaps the first character with the last character then second character with second last character and so on.

Mar 2008

- 1) Write a C++ program that will read a line of text and count the number of words in a text.
(Mar 2009, Mar 2012)

Oct 2008

- 1) Write a C++ program with computesphere() function that returns the volume v and the surface area s of a sphere with a given radius r.
- 2) Write a C++ program to read five elements of int array in reverse order and print the array i.e. read a[5] first and while printing print a[0] first.
- 3) Write a C++ program with average () function that returns the average of four input numbers. Use the following function prototype :
double ave (double x1, double x2, double x3, double x4) ;

Mar 2009

- 1) Write an object oriented program to implement a class convert to convert degree centigrade value to Fahrenheit degree value. [Hint $C = 5/9 (F - 32)$] where C is temperature in degree Celsius and F is temperature in Fahrenheit degree.
- 2) Write a program in C++ that inputs 10 numbers. Stores it in a one dimensional array and print the sum and average of all the elements of the array.

Oct 2009

- 1) Write a program in C++ to find sum of first 100 natural numbers.
- 2) Write a C++ program to input a word maximum length upto 15 characters from user and print each of its characters on a new line in a reverse order.
- 3) Write a program in C++ using OOP technique to find AREA of circle.

Mar-20	Explain friend function in C++ with example.
Mar-20	What is constructor and destructor? Explain each with the help of suitable example.
Mar-20	Define following terms in C++ file handling: i) ifstream ii) ofstream iii) fstream
Mar-20	Explain how a member function is defined outside class with example.
Mar-20	Write C++ declaration for the following : i) Array of 10 integers.ii) Pointer to character variable.iii) Object of the class test.
Mar-20	What is polymorphism? Explain how it is achieved by : i) Compile time ii) Runtime
Aug- 18	Explain basic type to class type conversion with suitable example.
Aug- 18	Describe how member functions are defined inside and outside a class in C++ ?
Aug- 18	List any six characteristics of virtual function.
Aug- 18	What is friend function? Write four rules for friend function.
Aug- 18	Explain in short three special characteristics of static data member function in a class.
Aug- 18	Write four difference points between Record and Linear array.
Aug- 18	Explain following functions related to file handling in C++. i) seekg () ii) tellg () iii) eof () iv) open ()
Mar- 18	Define OOP. Write its features.
Mar- 18	Explain how member functions of class can be defined outside the class definition and inside class definition with example in C++.
Mar- 18	Write the use of following file pointers with example: i) seekg () ii) seekp () iii) tellg () iv) tellp ()
Mar- 18	Explain Constructor and Destructor with example in C++.
Mar- 18	Write any eight basic rules for virtual function that satisfies the compiler requirements.
Mar- 18	Differentiate between Traditional Procedural Programming Approach and Object Oriented Programming Approach.
July -17	Explain the following OOP concepts : i) Data Hiding ii) Data Encapsulation iii) Data Abstraction
July -17	What is a Constructor? State any four syntax rules for writing constructor.
July -17	State any eight rules for operator overloading.
July -17	What is a Friend Function? Give any four characteristics of friend function.
July -17	What is Polymorphism? Explain the different types of Polymorphism with suitable examples.
July -17	What is a pointer in C++? Give the advantages using pointers.
Mar -17	What is friend function? Write any four characteristics of friend function.
Mar -17	Explain in short the three special characteristics of a static data member in a class.
Mar -17	What is Operator Function? Describe the syntax of an Operator Function.Explain the difference between Operator Function as the Member Function and Friend Function.
Mar -17	Explain different types of inheritance with suitable diagram.
Mar -17	Explain the concept of function overloading with example.
Mar -17	Explain the use of Scope resolution Operator and Memory Management Operators in C++ with example.
July-16	What do you mean by object based programming languages and object oriented programming languages? State the relationship between these languages.
July-16	What does inheritance mean in C++? Explain in brief with example any two forms of inheritance.
July-16	What do you mean by destructor in C++? What is the importance of destructor? State the order in which it destroys the objects.
July-16	What is friend function? State any four special characteristics of friend function.
July-16	What is operator overloading? What is an operator function? Describe the syntax of an operator function.
July-16	Explain with example the following functions used in C++ for file handling i) open () ii) close () iii) get () iv) put ()
Mar- 16	State any six characteristics of constructor.

Mar-16	What is polymorphism? Explain Compile Time and Run Time polymorphism
Mar-16	What is Operator Function? Explain difference between Operator Function as Member Function and as a Friend Function.
Mar-16	Explain Local Data and Global Data or variable in C++ using example.
Mar-16	State characteristics of Friend Function.
Mar-16	Using examples explain how files are opened and closed in C++. State any four file modes.
Mar-16	State three characteristics of Static Data.
Mar-16	Explain use of Memory Management Operators in C++.
Oct-15	List the different types of data types used in C++ with example.
Oct-15	What is constructor? Give an example.
Oct-15	Give two examples of declare member function: i) inside the class ii) outside the class
Oct-15	Explain library function : i) strcpy () ii) strcmp ()
Oct-15	Give any six rules of operator overloading.
Mar-15	Define the following terms in C++ : i) Data Abstraction ii) Operator Overloading iii) Data Encapsulation
Mar-15	What is Virtual Function in C++? Give any six rules to write Virtual Functions.
Mar-15	What is Function Overloading? Give examples of Function Overloading.
Mar-15	What is Constructor and Destructor in C++? Give example of Constructor and Destructor in a class.
Mar-15	What is inheritance? Explain any two types of inheritances with memory.
Oct-14	What is call by reference? Explain with suitable example.
Oct-14	With suitable example explain local and global variable in C++.
Oct-14	What is polymorphism? Give and define their types with example.
Oct-14	What is constructor? Give its syntax and two rules to define constructor.
Oct-14	What is array? How they are represented in memory? Explain with example.
Oct-14	What is a class? Give the basic structure to define the class with example.
Mar-14	What is class? How member functions defined inside and outside the class? Give example.
Mar-14	Explain Constructor and Destructor with the help of an example.
Mar-14	What is Inheritance? State different types of Inheritance with the help of diagrams.
Mar-14	What is pointer? Explain with the help of an example how address of a variable is accessed with the help of a pointer?
Mar-14	What is reference variable? Explain function call by reference with the help of an example.
Mar-14	Write declarations of the following : i) An array of 8. ii) An array of 8 pointers to floats. iii) Prototype of function that returns pointer to float (no parameters).
Mar-14	State any eight rules for Virtual Functions in C++.
Oct-13	State any six features of OOPs.
Oct-13	What is an operator overloading? What is an operator function? Describe the syntax of the operator function?
Oct-13	What is a constructor in C++? State any four special characteristics of constructor function.
Oct-13	Explain in short the three special characteristics of static data member in a class.
Oct-13	Explain in short the steps involved in process of operator overloading in C++.
Oct-13	State any eight basic rules for virtual function that satisfy the compiler requirement.
Mar-13	State any six principle advantages of Object Oriented Programming.
Mar-13	State any eight rules for operator overloading in C++.
Mar-13	What is class in C++? Write general form of class declaration.
Mar-13	What is constructor in C++? State any four special characteristics of constructor function.
Mar-13	Explain in short the steps involved in the Process of Operator Overloading in C++.
Mar-13	What does polymorphism means in C++? How is polymorphism achieved at compile time and runtime
Oct-12	Explain the following concept related to Object Oriented Programming: i) Class ii) Inheritance iii) Polymorphism

Oct-12	What is operator overloading? State any six rules for overloading the operator in C++.
Oct-12	Explain how the memory address of a variable can be accessed in C++.
Oct-12	What is constructor? What are the rules for writing the constructor.
Oct-12	Describe how the member function of a class can be defined outside the class definition and inside the class definition.
Oct-12	What is the function of each of the following file stream classes? i) ifstream ii) ofstream iii) filebuf iv) fstream
Mar-12	Explain Insertion and Extraction operators in C++.
Mar-12	Describe with an example, how member functions of a class can be defined outside the class definition and inside the class definition.
Mar-12	State any six principal advantages of Object Oriented Programming.
Mar-12	Explain difference between operator function as a member function and as a friend function.
Mar-12	What is a constructor and destructor? Give one example for each.
Mar-12	What are classes in C++ for tile stream operation? How do you open and close file in C++? Mention any four file modes.
Oct-11	Explain call by value and call by reference by giving an example for each.
Oct-11	What is operator overloading? State the three steps involved in the process of overloading an operator.
Oct-11	What is class? Give a general form of a class declaration.
Oct-11	What is inheritance? Mention the different types of inheritance with suitable Diagram.
Oct-11	What are different file mode parameters in C++?
Oct-11	Write any six features of Object Oriented Programming.
Oct-11	What are object? Describe how members of a class can be accessed using objects of that class.
Mar-11	Differentiate between Traditional Procedural Programming Approach and Object Oriented Programming Approach.
Mar-11	What is class? Explain general form of class declaration.
Mar-11	What is Pointer? What are advantages of pointer in C++?
Mar-11	What is constructor and destructor? State difference between them.
Mar-11	Explain in brief the three special characteristics of static data member in a class.
Mar-11	What is operator function? Describe syntax of an operator function. Explain difference between operator function as a member function and as a friend function.
Oct-10	Explain any six advantages of OOPs.
Oct-10	State any eight rules for operator overloading in C++.
Oct-10	What is constructor? With suitable example, state how a constructor is declared and defined.
Oct-10	What is friend function? State any four special characteristics of friend function.
Oct-10	What is operator overloading in C++? What is an operator function? Describe syntax of an operator function.
Oct-10	What is inheritance in C++? Explain in brief three different types of Inheritances with suitable diagrams.
Mar-10	What do you mean by Object Based Programming and Object Oriented Programming Languages. State relationship between them.
Mar-10	Explain in short the steps involved in the process of operator overloading in C++.
Mar-10	What is a destructor? What is the importance of a destructor? State order in which it destroys the object.
Mar-10	What is class in C++? Write general form of a class declaration.
Mar-10	Explain three types of data conversions in C++ with a suitable example:

H.S.C. board Probable Marks : 41

1 Mark	Objective	1	1X1	= 01
3 Mark	Question	4	4X3	= 12
4 Mark	Question	2	2X4	= 08
	5 Marks	C++ programs	4	5X4 = 20

Q) What is C++? What are advantages of C++?

C++ is an object oriented programming language. Initially C++ was named as “C with classes”. C++ was developed by Bjarne Stroustrup at AT and T Bell LAB, USA in early eighties.

Advantages/Features of C++ :

The C++ provides following advantages over any procedure oriented language such as C language:

- 1) C++ is an incremented version of C language. So it is called as superset of C language. Because all programs of C language can also in C++ language so it called superset/ incremented version of C language.
- 2) The important facilities added in C++ are classes, function overloading, operator overloading etc.
- 3) C++ allows user to creates an abstract data types which inherits properties from existing data types called as inheritance. So it supports reusability.
- 4) C++ supports polymorphism,
- 5) Any real life application such as editors, compiles, databases, communication systems etc can be built by C++.
- 6) Object oriented libraries can be built by C++.
- 7) C++ programs can be easily implemented, maintained and expanded.

Q) Differentiate between traditional procedural approach and object oriented programming approach.

Traditional Procedural Programming Approach	Object Oriented Programming Approach
1) In this approach the problem are viewed as a sequence of things to be done.	1) In this the problems are decomposed into number of entities called as objects and then builds data and functions around these entities.
2) Emphasis is doing things.	2) Emphasis is on the data rather than procedure.
3) Large programs are divided into smaller programs known as functions.	3) Programs are divided into number of runtime entities are called as objects.
4) Data are moves openly around the system from function to function.	4) Data are hidden and can not be accessed by an external functions.
5) Follows top-down approach in program design.	5) Follows bottom-up approach in program design.
6) Examples : C language, VB, COBOL, PASCAL	6) Examples : C++, Smalltalk, Ada

Q) What different data types in C++? Explain it detail?

The data type is required in every variable, function, array, string, pointer declarations. The data type name suggests that it decides the type of variable, function, array, string, pointer etc. It also decides how much space it reserves in memory. It also decides which data that variable, function, array, string, pointer etc holds during the program execution.

The data type in C++ is classified into three groups such as follows :

1) Built in type OR basic data type OR primary data type.

i) Integral type : -It includes integer and character data.

They are two types such as follows :

a) int : - An int type occupies 2 bytes in memory and holds integer constants i.e. whole numbers only.

b) char : The char type occupies 1 byte in memory and holds character data.

i) Floating type :-

It holds floating constants or real numbers. The constants contain decimal point numbers. They are two types such as follows:

a) float : - The float type occupies 4 bytes in memory and holds real constant i.e. constant have decimal point.

b) double :- The double type occupies 8 bytes in memory and holds real constants or fractional numbers.

ii) void type :- void data type is used in following ways :

a) To specify the return type of a function when it is not returning any value.

b) To indicate an empty argument list to a function.

c) To declare generic pointer.

2) Secondary data type OR Derived data type: These are derived from primary data type.

i) Array ii) Function iii) Pointer iv) String

3) User Defined data type OR Abstract type

C++ allows user to create a new abstract data types which can behave like any built in data type.

These are called as user defined data types. These include structure, union, class and enumeration etc.

i) Class ii) Structure and Union and Enumeration

SrNo	Data type	Bytes OR Size
1	signed char	1
2	unsigned char	1
3	short int OR signed int	2
4	unsigned int	2
5	Float	4
6	Double	8
7	long double	10

Q) Explain insertion and extraction operators in C++.

Insertion Operator : In C++ the operator << is called as insertion operator. It is used with cout i.e. output statement it is used to insert the content of the variable value or it inserts the enclosed string given in double quoted. It is also called as put to operator.

Example : 1) cout<<a ; This statement shows or output the value of a.

2) cout<<"\nThis is insertion operator" ; This statement shows or output given string enclosed in double quote.

Extraction Operator : In C++ the operator >> is called as extraction operator. It is used with cin i.e. input statement it is used to extract or inputs the value given at execution time of program and stores that constant value at that variable given in program.

Example : cin>>a ;

In above statement the value given at program execution i.e. at runtime is stored to variable a.

Q) Write a short note on scope resolution operator with example.

The operator :: is called as scope resolution operator. The scope resolution operator is used to access a global variable from a function in which a local variable is defined with same name as a global variable i.e. two variables have same name one is global variable and one is local variable.

Example :

```
int a = 20 ; // global variable which is declared and defined externally i.e. outside main function
void main ( )
{
    int a = 10 ; // local variable which is declared and defined internally and in scope of main ( )
    cout<<" Local a = "<<a ; // displays value of a is 10
    cout<<"Global a="<< : a ; //displays value of a is 20 because scope resolution is used.
}
```

Q) Explain memory management operators used in C++ ?

There are two types of memory management operators used in C++ is called as new and delete operators.

- 1) new : - It is a memory management operator used to allocate a memory in memory location. It is also used for dynamic storage allocation. This operator can be used to create object of any type.

Syntax : pointer variable = new data type ;

- 2) delete : - This operator is used for deallocation. This operator releases or free memory space in memory.

Syntax : delete variable ;

Example :

```
void main ( )
{
    int *a = new int ;
    *a = 100 ;
    cout<<"\nUsing new operator"<<*a ;
    delete a ;
}
```

Q) What is function in C++ write a suitable example.

A self contained block of statement or collection statements in its block is called as function. The function block is written separately i.e. defined separately from main () but before definition they are declared at outside the main function.

Terms related to functions :

- 1) Function prototype : -

This is a statement declared outside the main function i.e. above main () from this statement function are identified. This is also known as function declaration.

Syntax : return type functionname (type argument1, type argument2, ..., type argument n) ;

In above syntax return type of function is either char/int/float/double/void etc each arguments or parameters in function have its own data type either char/int/float/double etc.

- 2) Function definition :

This is a collection of statement of function. It tells that how function is implemented i.e. what function is do is defined in this block. The function definition block is defined outside main () .

- 3) Function call :

This is a statement is used in main () . The main () function calls subfunction is given in function call statement.

Example :

```
void add ( int a, int b ); // function prototype of declaration
```

```
void main ( )
```

```
{
    int a=10;
    int b = 20 ;
    cout<<"\nEnter two numbers";
    cin>>a>>b;
    add(a,b); // function call
    getch() ;
}
```

```
void add ( int a, int b ) // function definition
{
```

```
    int c ;
    c = a + b ;
    cout<<"\n"<<c;
}
```

Q) Explain the concept of function overloading with suitable example.

More than one function name having the same name but different task is called as function overloading. Overloading refers to the use of same thing for different purposes. It is also called as function polymorphism or compile time polymorphism.

In this C++ compiler finds more than one function have same name thus appropriate function is selected for execution in function call is dependent on number of arguments is given with function call, type of argument used in functions.

Thus in functions use different data types for arguments and number of arguments are different i.e same data type and same arguments are used in two functions then ambiguity is occurred thus no any functions are executed.

Example

```
int area ( int x ) ;
int area ( int a, int b );
void main ( )
{
    cout<<area ( 10 );
    cout<<area ( 2 ,4 );
}
int area ( int x )
{
    return (x * x ) ;
}
int area ( int a , int b )
{
    return ( a+b ) ;
}
```

In above example the function area () is overloaded. When a function is called then C++ compiler first matches the arguments given in call and its type. If match is found then appropriate function is selected for execution.

In above example both area () have different argument numbers one has one argument other has two arguments. This area () have same type argument as integer but number of arguments are different so overloading function are not ambiguous in execution. If two function have same argument type and same number of argument then overloading function can not execute function.

Q) What is default argument in C++ ? Write advantages of using default argument.

When a default value or constant value is assigned to a function in function declaration and definition but from right to left i.e. can not specify default argument at middle. Then that argument is called as default argument.

Example :

```
float area ( int r , float pi=3.14 ) ;
```

In above example float pi=3.14 is called default argument because the variable pi assigns a default or constant value 3.14.

Advantages of using default argument :

When default argument is used then it provides following advantages:

- 1) These are useful in situations where some arguments have same values.
- 2) It provides better flexibility to programmers by allowing to use particular arguments that are meaningful to particular solution.
- 3) Use default arguments to add new parameters to the existing functions.
- 4) Default arguments can be used to combine similar functions into a single function.

Q) Explain in detail inline function? What are different situations in which compiler may ignore inline request?

An inline function is a function that is expanded inline when it is invoked or initiated. The C++ compiler replaces function call with the corresponding function code. In inline function inline keyword is used with function name.

Example

```
inline int area ( int a, int b )
{
    return ( a*b ) ;
}
```

In above example inline keyword is used before function return type and function name.

Following are some situations in which compiler may ignore inline request :

- 1) For functions returning value then inline request is ignored.
- 2) The program contains loop statements, switch case statement or goto statements then inline request is ignored.
- 3) For function not returning value i.e. if a return statement is exists then inline request is ignored.
- 4) If function contains static variables then inline request is ignored.
- 5) If function contains recursion then inline request is ignored.

.Q) What is pointer? Write the advantages of using pointer?

A pointer is a variable which holds the memory address of other variable. The pointer operator * is used to declare pointer in C++ which is called value at pointer.

Example : int *a ;

In above example a is a pointer variable which is point to variable whose data type is integer. The data type of pointer a is not integer but data type of variable which pointer a will points is integer.

The pointer holds address of another variable is shown by as follows :

```
int *a, b;
*a = &b ;
```

In above *a=&b statement states that a is pointer variable who holds address of another variable b.

Advantages of using pointer such as follows :

- 1) Pointer allows to pass variables, arrays, functions, strings, structures, objects as function arguments.
- 2) Pointer allows to return structured variables from functions.
- 3) Pointer supports dynamic allocation and deallocation of memory segments.
- 4) By using a pointers, variables are swapped or interchanged without physically moving them.
- 5) Pointer allows to establish link between data elements or objects.

Q) Explain use of pointer variables for function definitions using call by value and call by reference.

OR Explain call by value and call by reference with one example of each.

The use of pointers in a function definition may be classified into two groups :

1) Call by value 2) Call by reference

1) Call by value :-When function is called in main () then program control is transferred from the main function to the calling function the main () transfers actual values that are copied to calling function. Within calling function i.e. sub function the actual values are changed or altered. Then when function is return from that function to main () back then the altered values or changed values in sub function can not transfers to main (). This mechanism is called call by value because main () calls calling function i.e. sub function by passing its actual values.

```
void add ( int a, int b ) ;
void main ( )
{
    int a=10 , b = 20;
    add ( a,b) ; // function call with call by value i.e. it passes actual value of a and b
    getch ( ) ;
}
void add ( int a, int b )
{ a =a + 10 ; // change values of in this function as a=20
  b= b + 20 ; // change value of b in this function as b= 40
  cout <<"a="<<a<<"b="<<b;
```

Thus in above example when program control is transferred from main () to calling function add () then main function passes values of a and b that are copied to calling function i.e. sub function. In calling function i.e. sub function the values are changed or altered but when it transferred back to main () function the change in sub function can not passed by sub function to main ().

2) Call by reference :-When function is called in main () then program control is transferred from the main function to the calling function the main () transfers address of the actual arguments that are copied to calling function. Within calling function i.e. sub function the actual values are changed or altered. Then when function is return from that function to main () back then the altered values or changed values in sub function can transfers to main (). This mechanism is called call by reference because main () calls calling function i.e. sub function by passing addresses or references of actual values.

```
void add ( int *a, int *b ) ;
void main ( )
{
    int a=10 ;
    int b = 20;
    add ( &a,&b) ; // function call with call by reference i.e. it passes address or reference
                  // of actual value of a and b
    getch ( ) ;
}
void add ( int *a, int *b )
{
    *a =*a + 10 ; // change values of in this function as a=20
    *b= *b + 20 ; // change value of b in this function as b= 40
    cout <<"a="<<*a<<"b="<<*b;
```

Thus in above example when program control is transferred from main () to calling function add ()

then main function passes values of a and b that are copied to calling function i.e. sub function. In calling function i.e. sub function the values are changed or altered but when it transferred back to main () function the change in sub function is passed by sub function to main ().

Q)What is array in C++?

An array is a collection of similar type data elements which are stored in consecutive or contiguous memory locations under common variable name. Array may be one dimensional or multidimensional.

Example : `int a [10]` // a is array of integer type holds 10 integer elements under common name

Generally C++ arrays are zero based. In above example the first array element has index 0 and it is referred as a [0]. Similarly second array element is a [1] and last i.e. 10th element is a [9]. Thus first element of array is index 0 and last element is size – 1. The subscript tells that variable is array and it have contain size of array.

Q) What is string in c++ ? Explain various string functions in C++ with example.

The one dimensional character array is called as string in C++. The string are used to manipulates text as such as words and sentences.

A string constant is 1D array of character terminated by null or \0.

Example : `char a [] = { "india" } ;` // where a is 1D character array is called string which holds string india.

Different string functions

The string functions in C++ are included or predefined into `string.h` header file. The string functions such as `strlen ()`, `strcpy ()`, `strcat ()`, `strcmp ()` and `strrev ()` etc. Thus to use this string functions `string.h` header file is included in program.

1) `strlen ()` This a string function counts the number of characters are present in string.

2) `strcpy ()` :- This a string function copies the contents of one string i.e. source string into other string i.e. target string.

3) `strcat ()` :- This a string function concatenates or combines the one string i.e. source string at the end of another string i.e. target string.

4) `strrev ()` :- This a string function that reverses contents of entered string.

Example:

```
#include <iostream.h>
#include<conio.h>
#include<string.h>
void main ( )
{
    clrscr ( ) ;
    char a [80] , b[80] ;
    cout<<"\nEnter string";
    cin.getline(a,80);
    cout<<strlen(a) ;
    cout<<strcpy(b,a) ;
    cout<<strcat(a,b) ;
    cout<<strrev(a) ;
    cout<<strcmp(a,b) ;
    getch ( ) ;
}
```

Q) What are different features provided by c-string in C++?

The string provides following features when it is used in program.

- 1) A string is a sequence of characters array with null terminated character or \0 is used to stores and manipulates the string.
- 2) For each character string in C++ inserts the null character or \0 automatically.
- 3) Each character in an array is occupied 1 byte of memory.
- 4) Elements of c-strings are stored in contiguous memory or consecutive memory locations.

Q) Explain the structure of a general C++ program.

OR Explain structure of object oriented C++ programming model.

Include Files
Class Declaration
Class Function Definition
Main Function Program

Fig Structure of C++ program

A typical C++ program contains four sections such as Include files, Class declaration, Class function definition and Main function program. These sections may be placed in different code files and then compiled independently or jointly.

1) Include Files: Any Input/Output functions are predefined into library subroutine files in header file. The all input/output functions such as cin and cout are predefined into iostream.h file. In this way console functions such as result get without screening and clear screen etc are defined in conio.h file. In this way some real life functions are predefined into respective header files. So to use this functions without rewriting its definition again include header files in program.

2) Class Declaration: The class is a abstract datatype. It begins with class keyword and same like as basic datatype class are created abstract datatype. The class declaration contains name of class and they have two members called as data member and member functions. The variables of class are called as data members while functions of class are called as member function.

3) Class Function Definition: The member functions in a class are called functions of class they performed given specific task is defined in its class function definition.

4) Main function Program : The actual execution of program is executed from here. It also executes class and its members.

Q) What is object oriented programming? Enlist the features of object oriented programming.

Definition of OOP:

An object oriented programming is defined as

“ Object oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand”.

Characteristics or Features of Object Oriented Programming:

1) Emphasis is on data rather than procedure. 2) Programs are divided into number of entities called as objects. 3) Data structures are designed such that they characterized the objects.

4) Functions that operates on the data of an object are tied together in the data structure.

5) Data is hidden and can not be accessed by external functions. 6) Objects may communicate with each other through functions. 7) New data and functions can be easily added wherever required.

Q) Explain the OOPS concept with an example of each.

The followings are some object oriented concepts these are used to develop object oriented programming :

1) Class :-

i) A class is a way to bind data and its associated functions together. ii) A class is a collection of objects of similar type. iii) A class is an abstract data type (ADT) or user defined data type in object oriented programming. Whose variable is object.

2) Object :-

i) An object is a runtime entity which is used at run time or dynamically. ii) To combine data and procedure a unit is formed which operates on that data. This unit is called as object. iii) When program is executed then object passes messages between them. iv) The object is declared and defined in main () function.

Example : void main ()

```
{
xyz d ; // xyz is a class and d is object class is data type and object is variable
d.print ( ) ;
```

 In above example object has dot operator which separates object name and member function. It is called as object identity operator or membership operator. The member functions of class are only accessible by object.

3) Data Abstraction :-

i) The collection of data and functions in a class is called as data abstraction. ii) The class binds data and associated functions together is called abstraction. iii) Abstraction refers to the act of representing essential features without including the background details of explanations. The class is a concept of abstraction.

4) Dynamic binding:-

i) The member functions are called at runtime only by an object in main function. This mechanism is called as dynamic binding. Class binds data and functions together while object accesses member functions of class. ii) This technique is associated with polymorphism and inheritance.

5) Data Encapsulation :-

i) The wrapping of data and code i.e. functions into a single unit i.e. class is called as data encapsulation. ii) Those functions are accessible which are wrapped in a class. An external function cannot access outside class is called data hiding.

6) Polymorphism :-

i) An ability to take more than one form is called as polymorphism. ii) There are two types compile time and runtime. iii) The compile time polymorphism is achieved by function overloading and operator overloading. iv) The runtime polymorphism is achieved by virtual function.

7) Inheritance :-

i) The mechanism of deriving a new class from base class is called as inheritance. ii) There are five types single, multiple, multilevel, hierarchical and hybrid.

8) Message Passing :-

i) This is a technique by which an object sends and receives messages one to another. ii) Message is a request for execution of procedure. iii) Object passes messages from one to another.

Q) What do you mean by object based programming and object oriented programming languages.

State relationship between them.

Object based programming language :-

It supports encapsulation and object identity without supporting important features of object oriented programming language such as polymorphism, inheritance and message based communication.

Example : Ada is object based programming language.

Object oriented programming language :-

It incorporates all the features of object based programming along with inheritance and polymorphism.

Example : C++ and Smalltalk are object oriented programming language.

Relationship between object based and object oriented programming language :-

The following expression tells the relationship between object based and object oriented programming language.

1) Object based programming language = encapsulation + object identity.

2) Object oriented programming language = object based features + Inheritance + Polymorphism

Q) What is class? Explain general form of class declaration.

Class is a way to bind data and its associated functions together. It allows the data and functions to be hidden if necessary from external use. When defining a class a new abstract data type that can be created that treated like any other built in data type.

Class specification or syntax of class :

Generally a class specification has two parts such as class declaration and class function definition.

1) Class Declaration: The class declaration describes the type and scope of its members.

Syntax or Form of Class Declaration:

```
class classname
{
    visibility label: Declaration of variables
    visibility label: Declaration of member functions
};
```

In above syntax

i) The keyword class specifies that what follows is an abstract data type class-name.

ii) The body of a class is enclosed within pair of braces (open and close) and is terminated by a semicolon.

The body of class contains declaration of class members such as:

a) Data Member of class :

The all variables declaration in class body are called as data members of class. They have its own visibility labels either public or private or protected. The members declared as private can be accessed only from within the class. It hides data from external use. The public data members can be accessed from outside the class also. **If visibility label are missed or not given then by default its visibility is private.**

b) Member function of class :

The all functions in body of class are called as member functions of class. They have its own visibility labels either public or private or protected. The members declared as private can be accessed only from within the class. It hides data from external use. The public data members can be accessed from outside the class also. **If visibility label are missed or not given then by default its visibility is private.**

2) Class function definition:

The class function definition describes how class functions are implemented. Class member functions are defined outside the class definition and inside the class definition.

Q) Describe how member functions of class can be defined outside the class definition and inside the class definition.

Member functions of class can be defined as two ways:

1) Inside the class definition :When a function is defined inside a class is called as inside class definition. It is treated same like inline function.

Syntax : class class name

```
{
    visibility label : return type function name ( )
    {
        // body of function
    }
};
```

Example :

```
class addition
{
    public: int a, b, c;
    public : void add ( )
    {
        c= a + b ;
        cout<<"\n Sum="<<c ;
    }
};
```

1) Outside the class definition:

When a function is defined outside the a class is called as outside definition. In this function is

accessed by scope resolution operator.

Syntax : class class name

```
{
    visibility label : return type function name ( ) ;
};
return type class name : : function-name (argument list )
{
    // body of function
}
```

Example :

```
class addition
{
    public: int a, b, c;
    public : void add ( ) ;
};
void addition : : add ( )
{
    c= a + b ;
    cout<<"\n Sum="<<c ;
}
```

Q) What is an object? Describe how members of a class be accessed using object of that class.

An object is a variable whose data type is class. In one class more than one object are possible to declares. The declaration of an object is similar to that of a variable of any basic type. The necessary memory space is allocated to an object at this stage. Each object of class are declared and defined only in main function.

Syntax : class-name object1-name, object2-name, ----- , object-n-name ;

Accessing members of a class using objects :

Each of member functions and data members are accessed by object. For this object uses dot operator between object name and function name. The dot operator or period connects the object name and the member function. The dot operator is also called as class member access operator.

Example

```
class addition
{
    public: int a, b, c;
    public : void add ( ) ;
};
void addition : : add ( )
{
    c= a + b ;
    cout<<"\n Sum="<<c ;
}
void main ( )
{
    addition x;
    x.add( ) ;
    getch();
}
```

Where addition x statement in main function is called as addition as class name is data type and x is object of class. The statement x.add() i.e. object x and dot operator separates object and member function. Thus object accesses member function of class and object must be declared and defined only in main function and accessed by data type class and variable object and dot operator with object and member function accesses it s function. Thus member functions of class are accessed by object.

The public variables of class can be accessed within main (). But private variables can not accessed inside main program they can be accessed by public functions of the same class.

Q) Differentiate between class and object?

Class	Object
1) It is a data structure that collects data members and member functions.	1) It is runtime entity of class or instance of class.
2) It is abstract data type in OOPS.	2) It is variable of class in OOPS.
3) The class is one structure.	3) There can be multiple objects of same type.
4) The class can not be used directly.	4) Object is the entity which can be used.
5) A class works like server.	5) Objects works like client.
6) Classes can be inherited.	6) Objects can be contained.

Q) Explain three special characteristics of a static data member and static member function in a class.

Characteristic/properties of Static data member;-

The three special characteristics of a static data member in a class are as follows:

- 1) It is initialized to zero when the first object of its class is created. No other initialization is permitted.
- 2) Only one copy of that member is created for the entire class and is shared by all the objects of that class, no matter how many objects are created.
- 3) It is visible only within the class but its life time is the entire program.

Characteristics/ Properties of Static member function :-

The following are characteristics of static member function:

- 1) A static function can have access to only other static functions which is declared in same class.
- 2) A static member function can be called using class name instead an object

Q) What is friend function? Write the characteristics of a friend function.

Those function which is declared in class body with keyword friend and which is friendly with more than one classes are called as friend function.

The non member functions of class can not access private i.e private data members and member functions can not accessed outside the class i.e external function can not accesses this. So to accesses private data members and member functions friend function is used.

Syntax :

```
class class-name
```

```
{
```

```
private : data members
```

```
private : ,member function
```

```
friend return-type function name(argument list) ; // friend function declaration
```

```
}; // class terminate
```

```
return-type friend function name ( ) //friend function definition can not contain class name  
and
```

```
friend keyword.
```

```
{
```

```
//body of friend function
```

```
}
```

Thus friend function is declared in class body is friend of more than one class and capable to accesses private data members and member functions outside the class by outside or external functions. But definition of friend function is outside the class body and it does not contains again class name because **friend function is declared inside the class but it is not member of class**. The definition of friend does not class name and keyword again friend.

Characteristics of friend function :-

The following are characteristics of friend function in C++

- 1) The friend function is not in the scope of the class i.e. it is not member of class which has been declared as friend keyword in class body.
- 2) Since it is not in scope of the class, it can not called by using object of that class . It is called like a normal C++ function.
- 3) It can be declared either in public or private part if a class without affecting its meaning.
- 4) Generally it has the objects as an arguments.
- 5) It can not accesses the member function directly and has to use an object name and dot operator with each member name.

Q) What is constructor? Why it is so called?

A constructor is a special member function of a class. Its task is to initialize the objects of its class. It is called special member because its name is same as that of the class to which it belongs. The constructor is initiated or invoked whenever an object of its associated class is created. It is called constructor because it constructs the values of data members of the class. A constructor can never return any value. Hence, it is written with no return type even void is not written.

Example:

```
class addition
{
    public : int a,b ;
    public : addition ( ) ; // constructor is declared name of class = name of constructor
};
addition : : addition ( )
{
    m=0 ;
    n = 1;
}
void main ( )
{
    addition x ; // object is created and it constructs data members
}
```

In above when object is created then constructor automatically initializes class data member. Thus in above object x not only creates the object x of type integer but also it initializes its data members as m=0 and n=1.

Q) What are the syntax rules for writing constructors?**OR Write/Enlist characteristics of a constructor function.**

The following are syntax rules to writing constructor these are also called as features or characteristics of constructor function.

- 1) The constructor name is same as the class name.
- 2) They do not have return types not even void and therefore they can not return values.
- 3) They can not be static or virtual. 4) They should be declared only in public section.
- 5) They can not be inherited though a derived class can call base class constructor.
- 6) Like other C++ functions they can have default arguments.
- 7) Constructor can not refer to its address.
- 8) An object with a constructor can not be used as a member of union.
- 9) They make implicit calls to the operator new and delete when memory allocation is required.
- 10) When a constructor is declared for a class then initialization of class objects become mandatory. Since constructor is invoked automatically when the objects are created.

Q) What are different forms of constructors in C++? OR Explain different types of constructors.

C++ have various ways to defines constructor such like as follows :

- 1) Parameterized Constructor : Those constructors can take one or more arguments or parameters are called as parameterized constructor.

Example : class fib

```
{
    int f0, f1 ;           // data members
    public: fib( int x , int y ) // parameterized constructor with two arguments
};
```

- 2) Non Parameterized Constructor : Those constructors can not take any arguments or parameters are called as non parameterized constructor.

Example : class fib

```
{
    int f0, f1 ;           // data members
    public: fib( ) // non parameterized constructor with no arguments
};
```

- 3) Default Constructor : Those constructors can take one or more arguments or parameters with default or constant value from right to left are called as default constructor or dynamic constructor.

Example : class fib

```
{
    int f0, f1 ;           // data members
    public: fib( int x , int y=2 ) // default constructor y argument contain default value 2.
};
```

- 4) Copy constructor :-Copy constructor are always used when the compiler has to creates a temporary object of a class object.

General form or syntax : class-name : : class-name (class-name &object-name)

Example:

fib : : fib (fib &x) // copy constructor

```
{
    f0 = x.f0 ;
    f1 = x.f1 ;
}
```

The copy constructors are used in following situations :

- a) The initialization of an object by another object of the same class.
- b) Return of object as by value parameters of a functions.
- c) Stating the object as by value parameters of a functions.

Q) How constructor are called?

The constructors called into two ways such as follows :

- 1) By calling the constructor explicitly :

The explicit call can be made as follows :

fib F = fib (0, 1) ;

Where fib is a class or data type and F is object of class or variable.

In fib(0,1) where fib is constructor with two arguments 0 and 1.

- 2) By calling the constructor implicitly :

The implicit call can be made as follows :

fib F (0,1)

In above fib is class or data type or constructor and F is object having two arguments.

Q) What is destructor? Write the syntax rules of destructor?

A destructor as name implies is used to destroy or delete the objects that have been created by a constructor. The destructor is invoked or initialized implicitly by the compiler upon exit from the program to clean up storage that is no longer accessible.

In other word destructor is defined as “a destructor function gets executed whenever an instance of the class i.e object of the class to which it belongs goes out of existence.

The destructor is started or begins with tilde sign (~) class name. e.g ~fib ().

Syntax rules for writing a destructor function:

- 1) A destructor function name is same as that of its class name. But it is preceded by a tilde sign(~) class name e.g. ~fib ().
- 2) It is declared with no return type since it can never return any value i.e. do not use even void.
- 3) It takes no arguments i.e. it does not contains any arguments or parameter.
- 4) It is public declared and defined.

Q) Differentiate between constructor and destructor.

Constructor	Destructor
1) constructs object.	1) Destruct object.
2) called explicitly.	2) called implicitly.
3) can take argument.	3) can not take arguments.
4) Initialized or called when object is created.	4) initialized or called when object goes outside scope.
5) can be overloaded.	5) can not overloaded.
6) They can not virtual.	6) They can be virtual.
7) Gets called in the sequence of an object are created.	7) Called in reverse order of creation of object.
8) class name = constructor name	8) class name = ~destructor name.

Q) What is operator overloading? Explain with suitable example? Write it s advantages.

OR Explain operator overloading with illustration. Write the advantages of operator overloading.

The mechanism to giving some special meaning to an operator is called as operator overloading. When operator is overloads then its original meaning is not changed

To define an additional task to an operator a special function is used is called as operator function. It is used to specify the relation of the operator to the class.

Advantages of operator overloading :

The followings are advantages of using operator overloading:

- 1) Operator overloading concept extends capability of operators to operates on user-defined data.
- 2) It can also be applied to data conversion.
- 3) Using operator overloading technique user defined data type behave in much the same way as the built in data type.

Example of operator overloading

Q) What are three steps involved in operator overloading.

The mechanism of giving special meaning to an operator is known as operator overloading.

To define an additional tasks to an operator a special function is used is called as operator function.

The process of overloading involves the following three steps:

Step 1 First create a class that defines the data type that is to be used in the overloading operations.

Step 2: Declare the operator function operator op () in the public part of the class. It may be either a member function or friend function.

Step3: Define the operator function to Implement the required operations.

Q) What is operator function? Describe the syntax of an operator function. Explain the difference between operator function as member function and friend function.

To define an additional task to an operator it specifies what it means in relation to the class to which the operator is applied. This is done with the help of a special function called as operator function which describes the task.

In other words “ A function which defines additional task to an operator or which gives a special meaning to an operator is called as operator function.

Syntax of operator function OR general form of operator function:

```
return-type class-name : : operator op ( argument list )
{
    // operator function body
}
```

Where return-type is type of value returned by the specified operation is int/char/float/double/void etc and op is the operator such as ++/- - /+/- etc is being overloaded.

Operator functions must be either member functions i.e. non-static or friend function.

Basic difference between operator function as a friend function and as a member function

Friend Function	Member function
1) only one argument for unary operators.	1) No arguments for unary operators.
2) Two arguments for binary operators.	2) one argument for binary operators.
3) Arguments may be passed either by value or by reference	3) Object is used to invoke the member function is passed implicitly and therefore is available for the member function.

Q) Write syntax rules for overloading operators.

There are certain restrictions and limitations for overloading operators. Some of them are listed below:

- 1) Only existing operators can be overloaded. New operators can not be created.
- 2) The overloaded operator must have atleast one operand that is user-defined data type.
- 3) The basic meaning of an operator can not be changed i.e. we can not redefine the plus(+) operator to subtract one value from the another.
- 4) The overloaded operators follow the syntax rules of original operators.
- 5) Following are some operators that can not be overloaded :

Operators	Operator Name
sizeof	Size of operator.
.	Membership operator.
.*	Pointer to member operator.
::	Scope resolution operator.
? :	Conditional operator.

- 6) Following certain operators can not overloaded using friend functions but member functions can be used to overload them.

Operators	Operator Name
=	Assignment operator.
()	Function call operator.
[]	Subscripting operator
→	Class member access operator

- 7) Unary operators overloaded by means of a member function take no explicit arguments and return no explicit values.
- 8) Unary operator overloaded by means of a friend function take one reference argument..
- 9) Binary operators overloaded through a member function take one explicit argument.
- 10) Binary operators overloaded through a friend function takes two explicit arguments.
- 11) When using binary operators overloaded through a member function the left hand operand must be an object of the relevant class.
- 12) Binary arithmetic operators such as +, -, *, and / must explicitly return a value. They must not attempt to change their own arguments.

Q) Enlist the operators which can not be overloaded and the operators where friend functions can not be used.

Operators which can not be overloaded as like follows :

Operators	Operator Name
sizeof	Size of operator.
.	Membership operator.
.*	Pointer to member operator.
::	Scope resolution operator.
? :	Conditional operator.

Operators where friend function can not be used

Operators	Operator Name
=	Assignment operator.
()	Function call operator.
[]	Subscripting operator
→	Class member access operator

Q) Write a short note on type conversions.

When constants and variables of different datatypes are mixed in an expression then C++ compiler applies automatic datatype conversions to the operands as per certain rules. The type of data to the right of an assignment operator is automatically converted to the type of variable on the left.

An assignment operator also causes the automatic type conversions.

Example:

```
int x; float y; y=29.13; x = y ;
```

In above example $x=y$ is type conversion statement it converts y to an integer before its value is applied to x . Thus the fractional part is truncated or removed because y is converted into integer value. So its value is 29.

Different types of data conversion in C++ :

The C++ provides three types of data conversions called as :

- 1) Conversion from built in type to class type. 2) Conversion from class type to built in type.
- 3) Conversion from one class type to another class type.

1) Conversion from built in type to class type :

In this data conversion constructor can be used for default type conversion from argument's type to the constructor's data type.

Example:

```
class time
{
    int hr, min;
public : time (int t) // constructor
{
    hr=t / 60 ; min = t % 60 ;
} };
void main ( )
{
    time z ; // z is object
    int duration = 90;
    z= duration ; // data conversion int to class type i.e built in type to class type.
```

2) Conversion from class type to built in type:

In this data conversion overloaded casting operator is used to convert a class type data to basic data type.

```
Syntax :      operator datatype ( )
{
                // function body
}
```

The conversion function must satisfy the following conditions :

- a) It must be a class member. b) It must not specify a return value. c) It must not have any arguments.

Example: `time :: operator int ()` // where `time` is class `operator int ()` operator function

```
{
    int min ; int min1 = hr * 60 ; min1 = min1 +min; return min1;
}
void main ( )
{
    time t ; // t is object class time
    int m = t ; // data conversion class to basic data type.
```

3) Conversion from one class type to another class type :

In this data conversion uses one argument constructor or conversion function depends upon the defining conversion routine in source class or destination class.

Example:

```
object a = object b
```

Where object a is a destination class object and object b is source class object. The constructor is placed in the destination class and conversion function is placed in source class.

Q) What is polymorphism? Explain runtime and compile time polymorphism.

OR What does polymorphism in C++ ? How is the same achieved at :

a) Compile time b) Runtime ?

Polymorphism means one name and multiple forms i.e an ability take multiple forms is called as Polymorphism.

Types of polymorphism :

There are two types of polymorphism such as

1) Compile time polymorphism :

In this information is known to the c++ compiler at the compile time and therefore the c++ compiler is able to select the appropriate function for a particular call at the compiler time itself This is known as compile time polymorphism.

Compile time polymorphism is also known as early binding or static binding. Because an object is bound to its function at compile time.

Example: Function overloading and operator overloading are called as compile time polymorphism.

In function overloading more than one functions having same name but different task. Thus appropriate function is selected by its type of argument, number of argument given etc. Thus in operator overloading one operator overloads other operator.

2) Runtime polymorphism :

In this the member function is to be selected or invoked while the program is running. So this is called as runtime polymorphism.

In this the appropriate member function can be selected at runtime and it is known as runtime polymorphism. To achieve this C++ supports mechanism of virtual function is used.

Example : Virtual function.

In this same function name in both base and derived class the function in base class is declared as virtual using keyword virtual. The c++ compiler determines which function to use at runtime based on the type of object pointed to by the base pointer.

Q) Write the difference in between static binding and dynamic binding with example.

Static Binding	Dynamic Binding
1) An object is bound to its function call at compile time.	1) Selection of the appropriate function is done dynamically at runtime.
2) The compiler knows the function information i.e. argument type, number of argument etc at the compile time itself. So it able to select appropriate function for a particular call also called as early binding.	2) The function is linked with a particular class much later after the compilation also known as late binding.
3) Example : Function overloading, Operator overloading	3) Example: Virtual function
4) class A { int x ; public: void show () { cout<<"\nBase Class" ; } } ; class B : public A { int y ; public : void show () { cout<<"\nDerived class"; } } ;	4) class A { int x ; public: virtual void show () { cout<<"\nBase Class" ; } } ; class B : public A { int y ; public : void show () { cout<<"\nDerived class"; } } ;

Q) Explain the concept of virtual function.

When user use the same function name in both the base and derived classes then the function in base class is declared as virtual using the keyword virtual.

When a function is made virtual then C++ compiler determines which function to use at runtime based on the type of object pointed to by the base pointer. Thus by making the base pointer to point two different objects it can execute different versions of the virtual function.

The virtual function can be accessed through the use of a pointer declared as a pointer to the base class. Since the prototypes of the base class version of a virtual function and all the derived class versions must be identical.

If two functions with the same name having different prototype then C++ compiler considers them as overloaded functions and the virtual function mechanism is ignored.

Q) Write basic rules for virtual function that satisfy the compiler requirements.

When virtual functions are created for implementing late binding then following basic rules are provided by C++ that satisfy compiler requirements:

- 1) The virtual functions must be members of base class.
- 2) They can not be static members.
- 3) They are accessed by using object pointers.
- 4) A virtual function can be a friend of another class.
- 5) A virtual in a base class must be defined even though it is not used.
- 6) The prototype of the base class version of virtual function and all derived class version must be identical. If two functions have different prototype then C++ compiler considers them as overloaded functions and not as virtual functions.
- 7) We can not have virtual constructors but we can have virtual destructors.
- 8) A base pointer can point to any type of derived object i.e. we can not use a pointer to derived class to access an object of the base type.
- 9) When base pointer points to derived class then incrementation and decrementation is only relative to its base type.
- 10) Virtual functions are defined in base class they need not be redefined in derived class.

Q) What is inheritance ? Explain with suitable example.

The mechanism of deriving a new class from an old one i.e existing class is called as inheritance. The old class is referred as base class and new class is referred as derived class.

Inheritance supports reusability. This is basically done by creating a new classes reusing the properties of the existing class. Functions and variables of this class has been tested can be used by object of another class. This is known as inheritance. Thus it saves execution time and code.

Syntax of inheritance :

```
class derived classname visibilityname base classname
{
    // members of derived class
};
```

Where visibility is optional if not given then by default private. If given may public/private.

Example

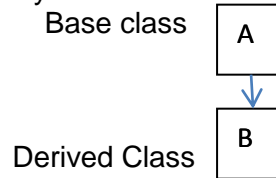
```
class base
{
    public void showbase ( )
    {
        cout<<" \nThis is the base class ";
    } } ; // Terminate base class
class derived: public base //declaration derived class as publically
{
    public : void showderived ( )
    {
        cout<<"\nIn derived class" ;
        showbase ( ) ; //base class function is used
    } } ; // Terminate derived class
```

Q) Explain different types of inheritances with suitable diagram.

There are five (5) type of inheritances in C++ such as follows :

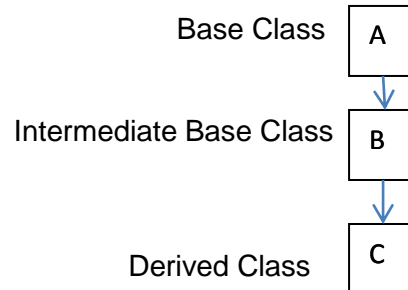
1) Single Inheritance :

A derived class with only one base class is called as single inheritance.



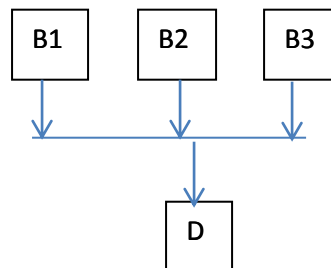
2) Multilevel Inheritance :

The mechanism of deriving one class from another derived class is multilevel inheritance.



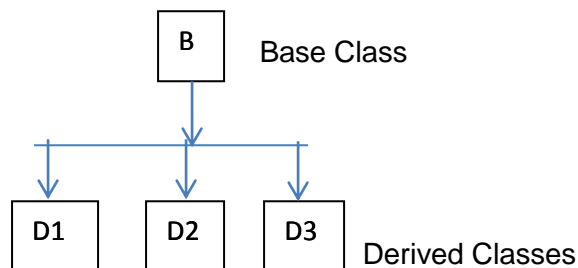
3) Multiple Inheritance :

When a class is derived from several base classes then it is called as multiple inheritance.



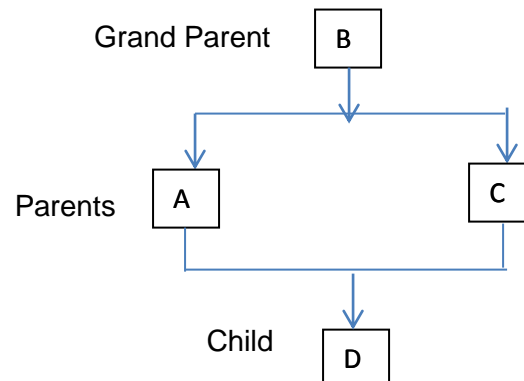
4) Hierarchical Inheritance :

When several derived class are derived from only one base class is known as hierarchical inheritance.



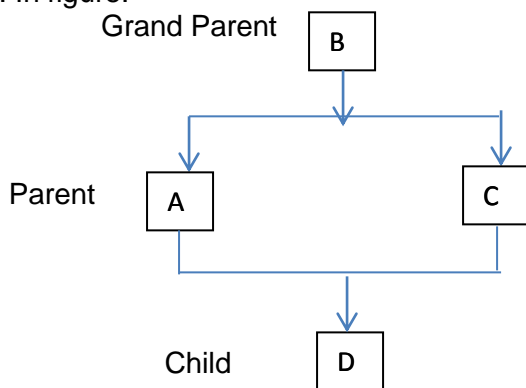
5) Hybrid Inheritance :

The inheritance which involves more than one inheritances is called as hybrid inheritance.



Q) What is virtual base class? Why is it necessary to define virtual base classes in some cases of hybrid inheritance?

Sometimes when hybrid inheritance is used then there are at least three levels as shown in following figure. In figure.



In figure class A and C are derived from class B and class D is derived from two parent class i.e. class A and class C. This means that class D may contains duplicate sets of members of class B i.e. the members of class B are inherited in class D twice via class A and via class C. This produces ambiguity. To avoid this ambiguity concept of virtual base class is used.

Thus duplication of inherited members due to multiple paths can be avoided by making the common base class as virtual base class while declaring the direct or intermediate base classes as follows:

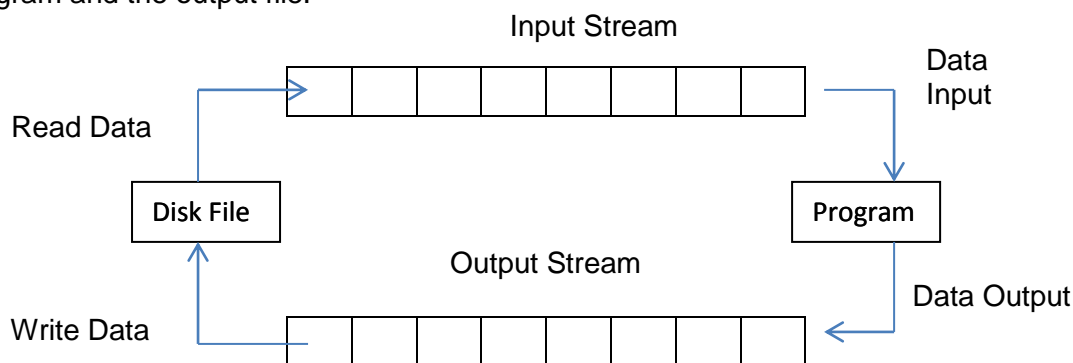
```
class B
{
}; // Grand Parent Class
class A: virtual public B
{
}; // parent1 class
class C : virtual public B
{
}; //parent 2 class
Class D : public A, public C //child class
{
}; // only one copy of B will be inherited.
```

When a class is made a virtual base class then C++ takes necessary care to see that only one copy of that class is inherited regardless of how many inheritance path exist between the virtual base class and the derived class.

Q) What are input and output streams?

The I/O system of C++ handles file operations which are very much similar to console input and output operations. It uses file streams as an interface between the programs and the files. The streams that supply data to program is known as input stream while the stream that receives data from the program is known as output stream.

In other words input stream extract or reads data from the file and the output stream inserts or writes data to the file. The input operation involves the creation of an input stream and linking it with the program and the input file. Similarly the output operation involves establishing an output stream and linking it to the program and the output file.



Q) Describe the various classes are available for file operations.

The I/O system of C++ contains a set of classes that defines the file handling methods. These includes ifstream, ofstream and fstream. These classes are derived from fstreambase class the corresponding iostream.h (header file) as shown in following figure. These classes are designed to manage the disk files are declared in fstream.h (header file) and therefore we must include this file in any program that uses files.

- 1) filebuf () : - Its purpose is to set the file buffers to read and write. It contains 'openport' constant and used in the 'open ()' of file stream classes. Also it contains close () and open () as members.
- 2) fstreambase () : - It provides operations common to the file streams. It serves as a base for fstream, ofstream and ifstream classes. It contains open () and close () functions.
- 3) ifstream:- It provides input operations and contains open () with default input mode. It contains functions such as get (), getline (), read (), seekg () and tellg () et c from ifstream class.
- 4) ofstream : - It provides output operations and contains open () with default output mode. It inherits functions such as put (), seekp (), tellp () and write () functions from ostream.
- 5) fstream :- It provides support for simultaneous input and output operations. It contains open () with default input mode. It inherits all the functions from istream and ostream classes through istream.

Q) What are classes in C++ for file stream operations? How do you open a file using open () function and close file using close ()?

Classes for file stream operation :-

The I/O system of C++ contains a set of classes that defines the file handling methods. They includes such as follows that are used as file stream classes for input/output file operations.

- 1) ifstream:- It provides input operations and contains open () with default input mode. It contains functions such as get (), getline (), read (), seekg () and tellg () et c from ifstream class.
- 2) ofstream :- It provides output operations and contains open () with default output mode. It inherits functions such as put (), seekp (), tellp () and write () functions from ostream.
- 3) fstream :- It provides support for simultaneous input and output operations. It contains open () with default input mode. It inherits all the functions from istream and ostream classes through istream.

The file stream classes such as ifstream, ofstream and fstream are derived from fstreambase and the corresponding iostream.h class. They are defined to manage the disk files and declared in fstream.h header file.

Open file using open () function : -

We can open files using open () by following two ways :

- 1) In first method open () function takes only one argument and that is file name.

Syntax

```
file-stream-class stream-object ;  
stream-object.open ( "file-name");
```

Example :

```
ofstream x . //where ofstream is output file stream class has stream object x.  
x.open ("c:\abc.cpp"); // one argument only name of file to open
```

- 2) In second method open () function takes two arguments such as file name and file open mode.

Syntax

```
file-stream-class stream-object ;  
stream-object.open ( "file-name", "file mode");
```

Example :

```
ofstream x . //where ofstream is output file stream class has stream object x.  
x.open ("c:\abc.cpp", ios::out); // two arguments file name and file open mode is output mode.
```

Closing a file using close () function :-

The close () is used to close a opened file

Syntax :

```
file-stream-class stream object ;  
stream-object.close ( )
```

Example:

```
ifstream x ;  
x.close ( ) ;
```

Thus close function have when ifstream class then close file in reading mode. While it have ofstream class then close file in write mode.

Q) What are different file modes for open a file ?

The following table lists the file mode parameters and their meanings is used with open ().

Parameter	Meaning
1) ios : : app	Appends to end-of-file.
2) ios : : ate	Goto end-of-file on opening.
3) ios : : binary	Binary file.
4) ios : : in	Open a file for reading only.
5) ios : : nocreate	Open fails if the file does not exist.
6) ios : : noreplace	Open fails if the file already exist.
7) ios : : out	Open file for writing only.
8) ios : : trunc	Delete contents of file if it exist.

Q) Explain in detail file pointers with their types and examples?

File pointer

Each file has two associated pointers known as the file pointers. Such as input file pointer and output file pointer.

Types of file pointer

The file pointer has two way to points the file called as input file pointer and output file pointer.

1) Input pointer OR get pointer : -

It is used for reading the content of a give file location.

Different input file pointers :

There are two input file pointers such as seekg () and tellg ().

- a) seekg () : - This is associated to ifstream class called as input pointer. It moves get pointer or input to the specific location is given in argument of function.

Example:

ifstream a ;

a.seekg (10) ; // the input file pointer forward or moves to 10 location form current position.

- b) tellg () : - This is associated to ifstream class called as input pointer. It gives or tells current position of the get pointer i.e input pointer.

Example :

ifstream a ;

a.tellg () ; // This input file pointer tells that current position of file pointer in file location.

2) Output pointer OR put pointer :-

It is used for writing to a given file location.

Different output file pointers :

There are two input file pointers such as seekg () and tellg ().

- a) seekp () : - This is associated to ofstream class called as output pointer. It moves put pointer or output to the specific location is given in argument of function.

Example:

ofstream a ;

a.seekp (10) ; // the output file pointer forward or moves to 10 location form current position.

- b) tellp () : - This is associated to ofstream class called as output pointer. It gives or tells current position of the put pointer i.e output pointer.

Example :

ofstream a ;

a.tellp () ; // This output file pointer tells that current position of file pointer in file location.

Topic Scope in H.S.C. Board Exam : 14 Marks
1 Mark Objective=1nos.=1X1=01
3 Mark Questions=1nos=3X1=03
5 Marks HTML programming questions=2nos=5X2=10

☞ What is HTML :

HTML stands for Hypertext Markup Language. HTML has its own syntax rules for communication. HTML is not really a programming language, but it is a markup language. Generally HTML is used for creating the web-pages. So it is a Internet Programming Language.

Using this language, programmer can create web-page which can be viewed in any web browsers, such as Netscape navigator or Internet Explorer. Hyper text is a ordinary text & adding extra features such as formatting, images, multimedia & links to other documents etc. The Mark-Up is the process of taking ordinary text & adding the extra symbols, such as editor's proof reading symbols are type of mark-up. Each of the symbol used for mark-up in HTML is command that tells the browser how to display the text or page. Thus a Mark-Up Language are special type of computer languages. They are concerned with only parts of documents according to their functions.

HTML is invented by Tim Berners-Lee at CERN in Europe LAB. It has extension for file is .htm or .html file.

☞ Advantages of HTML :

There are several advantages of HTML are enlisted below :

- 1) For creating HTML document, only text editor is needed. No special S/W is needed.
- 2) HTML document can be created on any H/W platform using any text editor.
- 3) HTML is easy to learn, use of implement.
- 4) Contains powerful formatting facilities.
- 5) Required HTML pages can be updated easily, without changing whole document.
- 6) Any HTML document can be traversed due to hyper linking facility is available.
- 7) Independent work can be done & needed not to worry about editing program.
- 8) If something is not working then finding error is easy in HTML.
- 9) HTML will not cost anything for its use. There are no expensive licenses to buy or no upgrades to purchase.
- 10) Learning HTML is very simple than any programming language.

☞ Disadvantages of HTML :

Apart from several advantages. HTML has certain limitations as listed below:

- 1) HTML is not programming language in true sense.
- 2) Any simple calculation cannot be done in HTML.
- 3) It can not be used to display even date.
- 4) The interactive web-pages can not build by HTML.
- 5) The web-pages developed in HTML can not behave like an application.
- 6) The web-pages developed in HTML do not have their own interface.
- 7) Hyperlink is provided in HTML, but for that we need a trip to server at each step.

☞ Features of HTML

The followings are some features provided by HTML

- 1) Hypertext Markup Language or HTML is a set of codes that is used to create documents and then it can be published on the world wide web.
- 2) HTML lets user to jump from topic rather than finding and reading information linearly.
- 3) Documents prepared in HTML includes reference graphic and formatting tags.
- 4) HTML is a hyperlink specification language.
- 5) HTML supports to frames including target windows and borderless frames.
- 6) It contains powerful formatting facilities for text, page, images etc.

Features of HTML

- 7) If defines the syntax and placement of special embedded directions which are not displayed by the browser but it tells the browser how to display the contents of the documents.
- 8) It supports for .BMP (paint files) and .GIF(graphical image file).
- 9) HTML supports forms which makes it possible to creates documents that collects and process uses input.
- 10) HTML tells how to makes a document interactive through special hypertext links.

Different Softwares that are used for writing HTML codes.

To develop or writes a HTML code the following softwares are used

- 1) Notepad or wordpad in windows accessories.
- 2) Microsoft Word in Microsoft office Suite.
- 3) Simple text in Macintosh.
- 4) Pico in Unix.

Structure of HTML web page or How HTML web page is developed or Syntax of HTML

Every HTML document has the same general structure and it consists of few tags that defines the page as a whole. The primary part of HTML document are denoted by <html>, <head> and <body> tags. Each of these tags are known as **Document Structure Tags**.

HTML files are always starts with <html> tag. Similarly ended with </html> tag. It declares text within web page viewed in a web browser.

Sections of HTML code syntax

HTML document can be divided into two sections as stated follows.

1) The head section:

It is like an introduction to the page. It generally consists of title of the page. To defines head add <head> tag at the beginning and </head> tag at end of headings.

2) The body section :

In this user enters the text images and other tags that will actually appears on the web page. To defines the body pace <body> tag is used at the beginning and </body> tag at the end after the head section.

The basic structure is like below

```
<html>
<head>
<title> title of the document </title>
</head>
<body>
Actual code of HTML is write here.
</body>
</html>
```

Procedure to prepare and view HTML document

To prepares HTML page following steps of procedure are followed.

- Step 1 Open a text editor e.g. Notepad.
- Step 2 Writes the appropriate HTML code.
- Step 3 Save the HTML code in a file having extension .html or .htm.
e.g. myhtmlweb.html
- Step 4 Open the browser where HTML page is opens e,g, Internet Explorer.
- Step5 Browse the file or file type in the appropriate address of the HTML file in the address bar to view the HTML page.

What are Tags :

A tag is a single unit of mark-up. It is a set of symbols, defined in HTML to have a special meaning. Every tag in HTML has a meaning and it is usually straight forward. Every tag starts with a less than sign (<) followed by a keyword & conclude with greater than (>) sign. These symbols together known as angle brackets. The attributes are inserted within tag that are used to affect tag's behaviors.

There are two types of tags in HTML :

- I) Start tag : Start tag are used to begin an effect.
- II) End tag : End tag are use to end that effect. Name of end tag is same as that of start tag but it is preceded by forward slash (/)

Essential tag OR Basic tag in HTML :

1) < HTML > tag : This tag tells browser that the file is opened in HTML file. This contains < HTML > start tag & < / HTML > end tag. The start tag starts opening contents & end tag ends.

e.g. < HTML >
 < HEAD >
 < TITLE > MY WEB BROWSER < /TITLE > < /HEAD >
 < BODY >
 < B > THIS IS MY WEB BROWSER CONTAINS HTML HEADING < /B >
 < /BODY >
 < /HTML >

2) < HEAD > tag : It defines the header area of the page which is not displayed within the page itself in the browser. It mainly contains < TITLE > tag definition. It also includes base, isindex, meta, script, style and link tags.

e.g. < HTML >
 < HEAD >
 < TITLE > MY WEB BROWSER < /TITLE > < /HEAD >

3) < TITLE > tag : It displayed the text between enclosed in start (< TITLE >) tag & end (< / TITLE >) on the title bar when executed within browser. It is placed in between < head > and end < /head > tag.

e.g. < HTML >
 < HEAD >
 < TITLE > MY WEB BROWSER < /TITLE > < /HEAD >

4) < BODY > tag : The actual contents of the web-page that will be displayed on the browser will appears in the body section of document. The body section starts with start (< BODY >) tag & end (< / BODY >)

Attributes of < body > tag : The < body > tag includes attributes such as background, bgcolor, text, link, vlink etc.

e.g. < html >
 < body background="c:\abc.bmp">
 < body bgcolor="red">
 < P > This is a web page which demonstrates the body tags explanation. < /P >
 < /BODY >
 < /HTML >

FORMATTING TAGS :

- 1) **< B > tag :** This tag is used for to bold text. The text appeared in the start **< B > tag** & end **** tag will displayed bold letters.
- ```
< HTML >
 < BODY>
 < B > This is a example of bold tag. < / B >
 < / BODY >
< / HTML >
```
- 2) **< I > tag :** This tag is used for to italics text. The text appeared in the start **< I > tag** & end **</ I>** tag will displayed italic letters.
- Example
- ```
< HTML >  
  < BODY>  
    < I > This is a example of italic tag. < / I >  
  < / BODY > < / HTML >
```
- 3) **< U > tag :** This tag is used for to underline text. The text appeared in the start **< U > tag** & end **</U>** tag will displayed underlined below letters.
- ```
< HTML >
 < BODY>
 < U > This is a example of italic tag. < / U >
 < / BODY >
< / HTML >
```
- 4) **< STRONG > tag :** This is a strong tag. The text appeared in the start **< STRONG > tag** & end **< / STRONG >** will displayed in bolded letters. But the main difference between **< STRONG >** & **< B >** is that the **< STRONG >** tag gives grammatical supports, it also supports pronunciation's to the text.
- e.g
- ```
< HTML >  
  < BODY>  
    < P > This is a best performance grammar < STRONG > using strong tag  
  < / STRONG >  
  < / P >  
< / BODY >  
< / HTML >
```
- 5) **< EM > tag :** This is a Emphasis tag. The text appearing in start **< EM >** & end **< / EM >** is displayed in the italic form. But the main difference between **< EM >** & **< I >** tags is that **< I >** only italics the word, whereas **< EM >** italics the word as full supports of grammars.
- e.g
- ```
< HTML >
 < BODY>
 < P > This is a best performance grammar < EM > using EM tag
 < / EM >
 < / P >
< / BODY >
< / HTML >
```
- 6) **< P > tag :** This is a paragraph tag. A paragraph can be created by enclosing text within paragraphs codes in start **< P > tag** & end **< / P > tag**.
- Attributes used with **< P > tag :**
- i) **< P align = "left" >** aligns to the left.
  - ii) **< P align = "right" >** aligns to the right.
  - iii) **< P align = "center" >** aligns to the center.
- e.g.
- ```
< HTML >  
  < BODY>  
    < P align = "center" > This is a paragraph tag.  
  < / P >  
< / BODY >  
< / HTML >
```
- Using this tag the text enclosed between this is paragraphed.
- ```
< / P >
< / BODY >
< / HTML >
```

7) < PRE > tag : This is a pre-format tag. The text appearing in the start < PRE > & end < / PRE > is displayed in the monospace form. Using this tag we can position the character. Generally this tag is used in the columnar lists.

e.g. < HTML >  
< BODY >  
< PRE > SrNO NameOfStudents RollNo Total < / PRE >  
< / BODY >  
< / HTML >

### **Line Break tag**

1) < BR > tag : This is a line-break tag. The text appearing in the start < BR > tag & end < / BR > tag writtens to the next line or new line. So it is also called as carriage return tag. It has no closing tag.

e.g. < HTML >  
< BODY >  
< B > This is a first line without using ( BR ) tag.  
< BR > This a second line using ( BR ) tag.  
< / BODY >  
< / HTML >

### **Ruler Tag :**

1) < HR > : This is a Horizontal Ruler Tag. The text appearing with start < HR > & end < / HR > tag divides separate sections by a ruler line.

Attributes used with < HR > tag :

- i) < HR SIZE > This tag is used for to sets the size of the horizontal ruler line.
  - ii) < HR WIDTH > This tag is used for to sets the width of the horizontal ruler line.
- e.g. < HR SIZE = “ 6 ” WIDTH = “ 6 0 % ” >

### **FONT TAGS :**

1) < FONT > This is a font tag. It is used for to sets the font name, font size & font colour.

Attributes used with < FONT > tag :

- i) < FONT FACE > : This used for to gives the name of font.
- e.g. < FONT FACE = “ ARIAL ” >
- ii) < FONT SIZE > : This is used for to gives the size of the font ranging size upto 1 to 7.

e.g. < FONT SIZE = “ 4 ” >  
< FONT SIZE = “ + 4 ” > Size makes larger 4 step to current font position.  
< FONT SIZE = “ - 4 ” > Size makes smaller 4 step to current font position.

- iii) < FONT COLOR > This is used for to gives the color to the font.

e.g. < FONT COLOR = “ PURPLE ” >

2) < SUB > tag : This is a subscript tag. The text appeared in the start < SUB > & end < / SUB > is displayed in the subscript form i.e. bit lower than text. It is generally used for to showing chemical formulae.

e.g. H < SUB > 2 < / SUB > O it shows H<sub>2</sub>O

3) < SUP > tag : This is a superscript tag. The text appeared in the start < SUP > & end < / SUP > is displayed in the superscript manner i.e. bit upper than the text. It is generally used to showing the mathematical formulae raised to the power.

e.g. 3 < SUP > 2 < / SUP > It shows 3<sup>2</sup>

4) < STRIKE > tag : This is a strike tag. The text appeared in the start < STRIKE > & end < / STRIKE > is displayed a line drawn through the middle of the text. The tag < S > is also used instead of < STRIKE > & < / S > is used instead of < / STRIKE >.

e.g. < STRIKE > This is a strike tag. < / STRIKE > is shown ~~This is a strike tag.~~

5) < BIG > tag : This is a big tag. The text appeared in the start < BIG > tag & end < / BIG > tag is displayed in the larger font. The < BIG > tag has also the same effect as < FONT SIZE = “ + 2 ” >. If already the size is largest then this tag is ignored.

e.g. < BIG > This is a effect of BIG tag < / BIG >

- 6)      < SMALL > tag :                      This is a small tag. The text appeared in the start  
< SMALL > tag & end    < / SMALL > tag is displayed in the smaller font. The  
< SMALL > tag has also the same effect as < FONT SIZE = “ - 2 ” >. If already the size is smallest then this  
tag is ignored.  
e.g.     < SMALL > This is a effect of small tag < / SMALL >
- 7)      < MARQUEE > tag : This is a marquee tag. The text enclosed in would rolled over horizontal line  
on the web-page. It is continuously rolling words till you close the application.  
e.g.     < MARQUEE > My marquee tag < / MARQUEE >
- 8)      < A > tag :    This is a anchor tag. It is used to creates link or hyper links. The text or images  
enclosed between starting tag < A > & ending tag < / A > is a link. This link is clickable in a graphical  
browser. The object to which the link has to be made is defined by the HREF attribute which refers to  
hypertext reference.  
e.g.     < A HREF = “ c:\abc.html ” > < / A >
- 9)      < ADDRESS > tag:    This tag displays web page’s information such as page URL ( uniform  
resource locator ). The text enclosed in the < ADDRESS > < / ADDRESS > is recognized by search engines  
as your address information. URL is address for a web-page. It is unique for each page.  
e.g.     http : //www.education.com/  
          In above example http is a hyper text transfer protocol & [www.education.com](http://www.education.com) is domain name i.e  
internet name of computer that is serving the information.  
e.g.     < ADDRESS > HTML NOTES email : [htmlnotes@hotmail.com](mailto:htmlnotes@hotmail.com) < / ADDRESS >

## ☞ List tag :

- 1) < LI > tag : This is a list tag. Generally used for collection group of items.  
e.g. <li> Sachin : It shows Sachin  
<li value= “10” > Sachin : It shows 10.Sachin

## ☞ Types of list tags :

There are two types of list tags used in HTML called as ordered list tag and unordered list tag.

### ☞ i) unordered list tag ( < UL > ) :

This is a unordered tag. The list are displayed in bulleted form. Unordered lists can be preceded by one of the several bullet styles like a closed circle (●), an open circle (○), a square (■). It contains the start < UL > & </UL>.

#### ☞ Attributes with < UL > tag :

- 1) < UL TYPE = “ CIRCLE ” : it used for a hollow bullet (○).
- 2) < UL TYPE = “ DISC ” : it used for a solid bullet (●)
- 3) < UL TYPE = “ SQUARE ” : it used for a square bullet (■)

e.g. < html>  
< body >  
< ul type = “ square ” >  
< li > Square bullet of unordered list.  
< ul type= “ disc ” >  
< li > Solid circle bullet of unordered list.  
< ul type = “ circle >  
< li > Open circle bullet of unordered list.  
</ul>  
</body>  
</html>

The above example shows as :

- Square bullet of unordered list.
- Solid circle bullet of unordered list.
- Open circle bullet of unordered list.

### ☞ ii) Ordered List tag ( < OL > ) :

Ordered List are numbered in some fashion. Ordered list can be preceded by Arabic Numerals, Uppercase or Lower Case Roman Numerals or Uppercase or Lowercase Alphanumeric characters. It contains start < OL > & end < / OL > tag.

#### ☞ Attributes used with ordered list tag:

- 1) < OL TYPE = “ 1 ” > : It shows Arabic numbers.

Eg. < OL TYPE = “ 1 ” >  
<LI> This is a first lie  
< LI> This is a second line.  
< LI > This is a third line.  
</OL>

It Shows as such like follows :

1. This is a first lie
2. This is a second line.
3. This is a third line.

2) < OL TYPE = “ a ” > : It shows lowercase alphanumeric.

Eg. < OL TYPE = “ a ” >  
<LI> This is a first lie  
< LI> This is a second line.  
< LI > This is a third line.  
</OL>

It Shows as such like follows :

- a) This is a first lie
- b) This is a second line.
- c) This is a third line.

3) < OL TYPE = “ A ” > : It shows uppercase alphanumeric.

Eg. < OL TYPE = “ A ” >  
<LI> This is a first lie  
< LI> This is a second line.  
< LI > This is a third line.  
</OL>

It Shows as such like follows :

- A. This is a first lie
- B. This is a second line.
- C. This is a third line.

4) < OL TYPE = “ i ” > : It shows lowercase roman numbers.

Eg. < OL TYPE = “ i ” >  
<LI> This is a first lie  
< LI> This is a second line.  
< LI > This is a third line.  
</OL>

It Shows as such like follows :

- i. This is a first lie
- ii. This is a second line.
- iii. This is a third line.

5) < OL TYPE = “ I ” > : It shows uppercase roman numbers.

Eg. < OL TYPE = “ I ” >  
<LI> This is a first lie  
< LI> This is a second line.  
< LI > This is a third line.  
</OL>

It Shows as such like follows :

- I. This is a first lie
- II. This is a second line.
- III. This is a third line.

6) < OL START = “ X ” > : X contains any starting number.

e.g < OL START = “10” > then list is ordered from 10.



### 👉 iii) Definition List tag ( < DL > ) :

The definition list elements uses < DL > start tag & </DL> end tag to creates a definition list. This list is rendered without bullets.

👉 < DT > tag is used for definition terms i.e. name or title of the item you are defining.

The 👉 < DD > tag is used for the definition themselves.

e.g. < DL >  
< DT > Term A  
< DD > Definition of term A  
< DT > Term B  
< DD > Definition of term B  
< /DL >

In above example the screen displayed

Term A  
Definition of term A  
Term B  
Definition of term B

### 👉 Heading Tag :

A web-page can have a maximum of six levels of headings.

- 1) < H1 > : It shows in very larger font than any other H2,H3,H4,H5 & H6.
- 2) < H2 > It shows in larger font than any other H3,H4,H5 & H6.
- 3) < H3 > It shows in larger font than any other H4,H5 & H6.
- 4) < H4 > It shows in larger font than any other H5 & H6.
- 5) < H5 > It shows in larger font than any other H6.
- 6) < H6 > It shows in very smaller font than any other H2,H3,H4,H5.

e.g. < H1 > This is a H1 heading tag </H2>

### 👉 Adding colors various colors in HTML page :

You adds different color schemes in HTML page. One of color scheme is known as RGB color scheme. It is a combination of RED, GREEN & BLACK colors & forms different colors. The colors are specified in your document by simple color name RED, GREEN, BLACK, YELLOW etc these are RGB schemes.

There are six body attributes : BGCOLOR, BACKGROUND, TEXT, LINK, VLINK & ALINK etc.

Eg. < BODY BGCOLOR= “ RED” TEXT = “ BLACK ” >  
< BODY VLINK= “ MARRON” >

## Adding Images in HTML :

You also insert pictures, images, photographs in HTML documents. These can be inserted by tag < IMG > & its many attributes.

< IMG > tag : This is an image tag. Its purpose is to include graphic images in the body of the web page. There are two types of images as listed below :

- 1) **Inline Images** : The inline images are aligned in top, middle and bottom. But by default it is middle of a line of text.
- 2) **Floating images** : These are two types aligned images as left and right. It causes text wrap around the image. To make an image as a separate paragraph, it is enclosed within paragraph elements as stated below :

< P > < IMG SRC = " C:\pic.bmp" > < / P >

## Attributes with < IMG > tag :

Generally SRC attribute & ALT attributes are used with < IMG > tag.

- 1) **SRC attributes** : It gives appropriate path of image file for searching.  
e.g < IMG SRC = " C:\pic.bmp" > This searches the file on C: & when located insert it.
- 2) **ALT attribute** : For any browser i.e. not displaying images, the alternate text contained inside the ALT attribute is displayed instead.  
e.g < IMG SRC = " C:\pic.bmp" ALT = " picture " >
- 3) **Images Alignments** : Images can be aligned with ALIGN attribute. It aligns images top, middle & bottom.  
Eg. < IMG SRC = " C:\pic.bmp " ALIGN = " MIDDLE" >  
< IMG SRC = " C:\pic.bmp " ALIGN = " TOP" >  
< IMG SRC = " C:\pic.bmp " ALIGN = " BOTTOM" >  
< IMG SRC = " C:\pic.bmp " ALIGN = " LEFT" >  
< IMG SRC = " C:\pic.bmp " ALIGN = " RIGHT" >  
< IMG SRC = " C:\pic.bmp " ALIGN = " CENTRE" >
- 4) **Setting images width & size** : Images width & size is also set by WIDTH & HEIGHT attributes.  
E.g < IMG SRC = " C:\pic.bmp " WIDTH = " 25 " HEIGHT = " 120 " >

## **Creating Hyper links in HTML documents :**

The hyper link name suggested that to links one document to another documents or to one image to another image or one address to another address etc. In HTML these links are also created by using a special tag called anchor tag <A>.

The links are created to jumps from one document to other or image or any local or web page by using tag <a>.

## **Attributes for <a> and <link>**

The following are the attributes appropriate to either anchor<a> or LINK elements.

- 1) HREF (links to object).
- 2) NAME (link from object).
- 3) TITLE(title of document).

Syntax :        < A HREF = “ c:\abc.html” > This links your main documents to abc.html document.

There are three types of hyperlinks in HTML as stated below:

### **1) Link to page on the WWW ( world wide web ) :**

e.g. < A HREF = <http://www.yahoo.com> > Enters your name < /A>

In this example the string “ Enters your name” is a hypertext link to the website indicated by URL ( universal resource locator ) specified.

### **2) Link to a image by image as a link :**

e.g < A HREF = “ c:\abc.bmp” > < IMG SRC = “ d:\xyz.bmp”></A>

In this example the two images are links together i.e. c:\abc.bmp is hypertext link to the image d:\xyz.bmp.

### **3) Link to document located in different directory :**

e.g < A HREF = “ d:\picture\ab.html” > Click here < /A>

In this example the link is created on the text string “ Clock here” when clicking on that string the destination web page is displayed which is specified in the path given

👉 **Creating tables in HTML page :** Tables are created in HTML page by a < TABLE > tag.

👉 1) < TABLE > This is a table tag. It organizes given data in a table format such as row & column format. Table tag is created by start < TABLE > & end < / TABLE > tag.

👉 **Attributes of < TABLE > tags :**

Following are attributes which uses within < TABLE > tag.

- a) < TABLE WIDTH > It specifies the width of the table.  
e.g. < TABLE WIDTH = " 50 % " > It specifies table width size is 50%.
- b) < TABLE HEIGHT > It specifies the height of the table.  
e.g. < TABLE WIDTH = " 50 % " TABLE HEIGHT = " 12 " > It specifies table width size is 50% & height of table is 12.
- c) < TABLE ALIGN = " LEFT " > It horizontally aligns left to the specified table. It is default.
- d) < TABLE ALIGN = " RIGHT " > It horizontally aligns right to the specified table.
- e) < TABLE ALIGN = " CENTER " > It horizontally aligns center to the specified table.
- f) < TABLE BORDER = " 5 " > It sets table border size is 5.
- g) < TABLE BORDER = " 1 " > It sets table border size is 1.
- h) < TABLE BORDER = " 0 " > It sets no border for table.

Note : By default table border is zero when not specified it considers zero.

- i) < TABLE CELSPACING = " 12 " > It creates 12 spaces between each cell.  
The cell spacing is not default you specifies any space size.
- j) < TABLE CELLPADDING = " 5 " > It increases the amount of spaces between a cell boarder & the cell data. In example amount space is 5. It is not default size you specifies any amount of size. When not specifies any cell space by default 1.
- k) < TABLE BGCOLOR > It sets the background color to table.

👉 2) < CAPTION > tag : This tag is used to crates a caption on the top of or below the table. It requires start < CAPTION > tag & end < / CAPTION > tag. It is aligned with align attributes. It aligns caption left, right, top & bottom as stated following example.

e.g < TABLE >  
    < CAPTION ALIGN = " TOP " >

It was table caption aligned top tag < /CAPTION >  
< /TABLE >

👉 3) < TR > It is a table row tag. It contains the start < TR > & end < /TR > tag. It creates a horizontal row of cells & contains table headings or table data. Each use of a table row i.e. < TR > begins a new table row. A row must contains atleast one table data element or table heading elements. Table data is determined by < TD > tag & table heading elements is determined by < TH > tag.

**It includes attributes like ALIGN, BGCOLOR & VALIGN.**

**HORIZONTAL ALIGN attribute with < TR > :** With < TR > tag we also uses horizontal align attributes such as left, right, center etc. Which aligns table row left, right, center respectively.

- e.g. < TR ALIGN = " LEFT " > It displays table row horizontally aligns to left.  
    < TR ALIGN = " RIGHT " > It displays table row horizontally aligns to right.  
    < TR ALIGN = " CENTER " > It displays table row horizontally aligns to center.

**VERTICAL ALIGN attribute with < TR > :** With < TR > tag we also uses vertical align attributes such as top, middle, bottom, baseline, etc.

- eg. < TR VALIGN = " TOP " > It displays table row vertically aligns to top side.  
    < TR VALIGN = " MIDDLE " > It displays table row vertically aligns to middle side.  
    < TR VALIGN = " BOTTOM " > It displays table row vertically aligns to bottom side.  
    < TR VALIGN = " BASELINE " > It displays table row vertically aligns to baseline side.

**COLOR attribute with < TR > :** We also specifies the background color to the table row. The attribute BGCOLOR is used with < TR > tag.

e.g. < TR BGCOLOR = " PURPLE"> It displays purple as background color for table row.

☞ **4) < TH > tag :** This is a table heading tag. It contains start tag < TH > & an end tag < /TH >. It creates headings. By default, text enclosed in start & end tag is bolded & centered form.

☞ **Attributes used with < TH > tag:**

1) **Width attribute** : This specifies the width of table heading.

e.g < TH WIDTH = " 50 " > It determines the width of table heading is 50.

2) **HEIGHT attribute** : This specifies the height of the table heading.

e.g < TH HEIGHT = " 30 " > It determines the height of table heading is 30.

3) **HORIZONTAL ALIGN attribute with < TH > :** With < TH > tag we also uses horizontal align attributes such as left, right, center etc. Which aligns table row left, right, center respectively.

e.g. < TH ALIGN = " LEFT " > It displays table heading horizontally aligns to left.

< TH ALIGN = " RIGHT " > It displays table heading horizontally aligns to right.

< TH ALIGN = " CENTER " > It displays table heading horizontally aligns to center.

4) **VERTICAL ALIGN attribute with < TH > :** With < TH > tag we also uses vertical align attributes such as top, middle, bottom, baseline, etc.

eg. < TH VALIGN = " TOP " > It displays table heading vertically aligns to top side.

< TH VALIGN = " MIDDLE " > It displays table heading vertically aligns to middle side.

< TH VALIGN = " BOTTOM " > It displays table heading vertically aligns to bottom side. < TH VALIGN = " BASELINE " > It displays table heading vertically aligns to baseline side.

☞ **COLOR attribute with < TH > :** We also specifies the background color to the table heading. The attribute BGCOLOR is used with < TH > tag.

e.g. < TH BGCOLOR = " PURPLE"> It displays purple as background color for table heading.

☞ **5) < TD > tag :** This is a table data tag. It contains start tag < TD > & an end tag < /TD >. It creates individual cells. The number of cells in the row determines the number of columns. By default, text enclosed in start & end tag is left & centered vertically.

☞ **Attributes used with < TD > tag:**

1) **Width attribute** : This specifies the width of table data.

e.g < TD WIDTH = " 50 " > It determines the width of table data is 50.

2) **HEIGHT attribute** : This specifies the height of the table data.

e.g < TD HEIGHT = " 30 " > It determines the height of table data is 30.

3) **HORIZONTAL ALIGN attribute with < TD > :** With < TD > tag we also uses horizontal align attributes such as left, right, center etc. Which aligns table row left, right, center respectively.

e.g. < TD ALIGN = " LEFT " > It displays table data horizontally aligns to left.

< TD ALIGN = " RIGHT " > It displays table data horizontally aligns to right.

< TD ALIGN = " CENTER " > It displays table data horizontally aligns to center.

4) **VERTICAL ALIGN attribute with < TD > :** With < TD > tag we also uses vertical align attributes such as top, middle, bottom, baseline, etc.

eg. < TD VALIGN = " TOP " > It displays table data vertically aligns to top side.

< TD VALIGN = " MIDDLE " > It displays table data vertically aligns to middle side.

< TD VALIGN = " BOTTOM " > It displays table data vertically aligns to bottom side. < TD VALIGN = " BASELINE " > It displays table data vertically aligns to baseline side.

☞ **COLOR attribute with < TD > :** We also specifies the background color to the table data. The attribute BGCOLOR is used with < TD > tag.

e.g. < TD BGCOLOR = " PURPLE"> It displays purple as background color for table data.

👉 6) **COLSPAN & ROWSPAN attributes used with < TD> & < TH :**

COLSPAN & ROWSPAN are the special attributes that can be used with < TD> & <TH> tags. There may be some situation, in which one cell of table to span more than one row or more than one column. For this merging situations, COLSPAN & ROWSPAN attributes are used with the < TD> & < TH > tags.

👉 **COLSPAN attribute :** This spans the relevant columns i.e it makes the cell contents merge to the another cells.

e.g < TD COLSPAN = “ 2 ” > This merges the two columns in table’s data elements.

< TH COLSPAN = “ 3 ” > This merges three columns for table heading.

👉 **ROWSPAN attribute :** This spans the relevant rows i.e it makes the cell contents merge to the another cells. It also specifies how many rows a cell should takes up.

e.g < TD ROWSPAN = “ 2 ” > This merges the two rows in table’s data elements.

< TH ROWSPAN = “ 3 ” > This merges three rows for table heading.

## **Study of VB Script :**

A script name suggested that it contains series of commands in its own script language tag. This script commands are executed under the host environment or server. Scripting languages are special programming languages. These are used on web-page to controls different elements of the page including controls frames & browser interface. Using scripts in web-page it runs dynamically Or actively this HTML page is then known as DHTML i.e. dynamic hyper text markup language.

In market most scripting languages are available but most popular scripting languages are used in creating web-page are VB Script & Java Script. VB script are writtens under the tag < SRIPT > tag.

## **\*\* Advantages or Features of VB SCRIPT :**


- 1) VB Script are designed to be faster. Because this script does not supports any strict data type i.e integer, character, float etc.
- 2) VB Script only contains one data type as varient.
- 3) VB Script provides safe security. It provides file system components which consists of objects, which can be used to perform I/O.
- 4) VB Script contains rich set of functions for performing various types of operations via array, manipulating functions, data & time functions, string functions, mathematical functions, conversion functions etc.
- 5) VB Script allows user to access & handles error through the use of Err object.

## **\*\* Disadvantages OR Limitations of VB Script :**

- 1) All VB Script arrays are zero based. It do not allows user to change the base of an array variables for specific implementation.

e.g. If we want to create array of 10 size numbers then array is started from 1. But we need to start array from 0.

- 2) VB Script can not creates user-defined data types.
- 3) VB Script can not supports Dynamic Data Exchange ( D.D.E.) .

 **< SCRIPT> tag :** The VB Script can be placed in < SCRIPT > start tag & </SCRIPT> end tag. It contains attributes LANGUAGE= “ VB Script “ as stated follows :

```
< SCRIPT LANGUAGE = “ VB Script ” >
// Scripting code
< /SCRIPT>
```

When browser finds the < SCRIPT> tag then it calls the VB Script interpreter to compile & executes the given code under < SCRIPT > tag.

e.g. // VB Script displaying local time & day.

```
< HTML >
< HEAD >
< TITLE > Local time & day < /TITLE >
< /HEAD >
< BODY >
< SCRIPT LANGUAGE = “ VB SCRIPT ” >
DOCUMENT.WRITE Time() & “ on” Month Name (Month(Now) , False)
& Day (Now) And Year (Now)
< /SCRIPT >
< /BODY >
< /HTML >
```

In above program when browser finds the < SCRIPT LANGUAGE = “ VB SCRIPT>

Tag then it calls interpreter to compile & execute code. The code of VB script is then placed to Time ( ) then it reads the local time, month name, day & year etc. After completion of procedure Time ( ) function then DOCUMENT.WRITE which writes the desired result.

## 👉 RGB codes or Color codes in HTML with RGB color coding system

Colors in HTML can be specified by color name or by color code known as RGB code. In RGB code R stands for red, G stands for green and B stands for Blue.

There are 16 color names in HTML such as black, silver, gray, white, maroon, red, purple, fuchsia, green, lime, olive, yellow, navy blue, teal and aqua.

RGB codes are always 6 numbers. First two numbers specify amount of red color. Next two numbers are specifies amount of green and last two numbers are specifies amount of blue color. By mixing these three primary colors in different amount it is possible to derives any color. RGB code uses hexadecimal numbering system.

Examples RGB codes

- 1) #FF0000 This is a RGB code for red color.
- 2) #0000FF This is a RGB code for blue color.
- 3) #00FF00 This is a RGB code for green color.

## HTML Colors - RGB Values

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

**Note :** All the browsers does not support rgb() property of color so it is recommended not to use it.

Following is a list to show few colors using RGB values.

| Color Name | RGB Color code in HEX | RGB Color Values |
|------------|-----------------------|------------------|
| Black      | #000000               | rgb(0,0,0)       |
| Red        | #FF0000               | rgb(255,0,0)     |
| Green      | #00FF00               | rgb(0,255,0)     |
| Blue       | #0000FF               | rgb(0,0,255)     |
| Yellow     | #FFFF00               | rgb(255,255,0)   |
| Aqua       | #00FFFF               | rgb(0,255,255)   |
| Fuchsia    | #FF00FF               | rgb(255,0,255)   |
| Silver     | #C0C0C0               | rgb(192,192,192) |
| White      | #FFFFFF               | rgb(255,255,255) |
| Gray       | #808080               | rgb(128,128,128) |
| Maroon     | #800000               | rgb(128,0,0)     |
| Purple     | #800080               | rgb(128,0,128)   |
| Lime       | #00FF00               | rgb(0,255,0)     |
| Olive      | #808000               | rgb(128,128,00)  |
| Navy Blue  | #000080               | rgb(0,0,128)     |
| Teal       | #008080               | rgb(0,128,128)   |



## H.S.C. Board Question Bank Objectives of HTML

- 1) To place the image into an HTML file ----- attribute is used in IMG tag.  
a) <url>      b) <alt>      c) <src>      d) <href>      (Oct 2003, Mar2006, Mar2007, Mar2011)

Ans c) <src>

- 2) VBScript can be executed in ---- web browser.      (Mar2004)  
a) Netscape navigator      b) Internet Explorer      c) Both      d) none of these

Ans c) both

- 3) HTML stands for -----      (Oct2004, Oct 2005, Mar2013)  
a) Hypertext Markup Language      b) Hightext Manipulation Language

c) Hypertext Manipulating Language      d) Hightext Markup Language

Ans a) Hypertext Markup Language

- 4) The long form of SGML is -----      (Mar2005)  
a) Standard Global Machine Language      b) Special Global Markup Language

c) Symbolic Generalized Machine Language      d) Standard Generalized Markup Language

Ans d) Standard Generalized Markup Language

- 5) ----- is name of the web browser.      (Oct2006)  
a) Embedded system      b) Netscape navigator      c) Oracle      d) C++

Ans b) Netscape navigator

- 6) COLSPAN attribute is used with ----- tag.      (Oct2007)  
a) <body>      b) <html>      c) <title>      d) <table>

Ans d) <table>

- 7) <A> tag has attribute ----- which defines URL of the document to be linked.      (Mar 2008)  
a) SRC      b) HREF      c) VREF      d) REF

Ans b) HREF

- 8) In HTML ----- attribute defines the name of the file in which the image is to be found.      (Oct2008)  
a) ALIGN      b) SIZE      c) SRC      d) BGCOLOR

Ans c) SRC

- 9) ----- tag is used to scroll the text.      (Mar2009)  
a) <STRIKE>      b) <MARQUEE>      c) <HR>      d) none of these

Ans b) <MARQUEE>

- 10) In HTML ----- tag is used for super script.      (Oct 2002, Oct 2009)  
a) <S>      b) <Super>      c) <sup>      d) <script>

Ans c) <sup>

- 11) ALIGN is not an attribute used with ----- <tag>      (Mar 2010)  
a) <body>      b) <hr>      c) <tr>      d) <table>

Ans a) <body>

- 12) ----- is a tag in HTML.      (Oct2010)  
a) ALT      b) SRC      c) IMG      d) ALIGN

Ans c) IMG

- 13) ----- tag is used for subscript in HTML code.      (Oct2011)  
a) <sup>      b) <subscript>      c) <sub>      d) <super>

Ans c) <sub>

- 14) ----- tag is used to put a horizontal ruler in HTML code.      (Mar2012)  
a) <HR>      b) <BR>      c) <P>      d) <TD>

Ans. a) <HR>

- 15) For green color RGB code is -----      (Oct2012)  
a) #FF0000      b) #0000FF      c) #00FF00      d) #00FFFF

Ans. c) #00FF00

- 16) The longform of HREF is -----      (Mar2002)  
a) Horizontal reference      b) Hypertext reference      c) Hyperlink reference      d) Hypermedia reference

Ans. b) Hypertext reference

- 17) ----- tag is used to put a line break in HTML code.      (Mar2003)  
a) <HR>      b) <BR>      c) <P>      d) <TT>

Ans b) <BR>

**H.S.C. Board Question Bank of HTML**  
**3 Marks Questions in H.S.C. BOARD Exam of HTML**

---

- 1) What is HTML? State the advantages and disadvantages of HTML.  
(Oct2003,Mar2006,Mar2009,Oct2009)
- 2) Explain the following HTML tags with one example of each :  
a) <PRE>    b) <MARQUEE>    c) <SUB> (Mar2004)
- 3) Explain the following HTML tags with one example of each :  
a) <BR>    b) <HREF>    c) <SUB> (Oct2004)
- 4) Explain the following HTML tags with one example of each :  
a) <OL>    b) <TR>    c) <B> (Mar2005)
- 5) Explain the following HTML tags with one example of each :  
a) <SUB>    b) <P>    c) <BODY> (Oct2005)
- 6) Explain the following HTML tags with one example of each :  
a) <HR>    b) <UL>    c) <PRE> (Oct2006)
- 7) Explain the following HTML tags with one example of each :  
a) <P>    b) <LI>    c) <PRE> (Mar2007)
- 8) Explain the following HTML tags with one example of each :  
a) <SUB>    b) <U>    c) <LI> (Oct2007)
- 9) Explain the following HTML tags with one example of each :  
a) <MARQUEE>    b) <SUB>    c) <BODY> (Mar2008)
- 10) Explain the attributes BGCOLOR and BACKGROUND of <BODY> tag. (Oct2008)
- 11) Explain use of the following attributes of <IMG> tag with suitable examples:  
a) ALT    b) HEIGHT    c) SRC (Mar2010)
- 12) Explain the following HTML tags with one example of each :  
a) <SMALL>    b) <BIG>    c) <SUP> (Oct2010)
- 13) Explain the following HTML tags with one example of each :  
a) <P>    b) <BR>    c) <HR> (Mar2011)
- 14) Give structure of HTML web page. (Oct2011)
- 15) Explain the following HTML tags with one example of each :  
a) <PRE>    b) <BR>    c) <SUB> (Mar2012)
- 16) Explain the following HTML tags with one example of each :  
a) <HEAD>    b) <TITLE>    c) <BODY> (Oct2012)
- 17) Explain the following HTML tags with one example of each :  
a) <I>    b) <LI>    c) <TH> (Mar2013)

**H.S.C. Board Question Bank of HTML programming**  
**5 Marks Questions in H.S.C. BOARD Exam of HTML**

- 1) Write the exact output of the following HTML code with font specifications in brackets. (Oct2003)

```
<html>
<title> Introduction</title>
<body>
<h1>COMPUTER SCIENCE </h1>
<hr>
<u> SCHAUM’S SERIES </u>
<hr>
<u> SCHAUM’S OUTLINE SERIES </u>
<hr>
<h5>SEYMOUR LIPSCHUTZ</h5>
</body>
</html>
```

- 2) Write the HTML code for the following (Oct2003)

SCIENCE		
F..Y.B.Sc.	S.Y.B.Sc.	T.Y.B.Sc.
300	100	25
ARTS		
F..Y.B.A.	S.Y.B.A.	T.Y.B.A.
200	150	40
COMMERCE		
F..Y.Bcom	S.Y.B.Com	T.Y.B.Com.
300	70	50

- 3) Write the HTML code for the following table : (Mar2004)

		Year		
		1999	2000	2001
Sales	Units	300	750	1200
	Income	Rs. 3000	Rs. 7500	Rs. 12000

- 4) Write a HTML code using VB Script for designing a Web-Page which greets “Good Morning” if time is from 12:00 am upto 12:00pm else it greets “Good Afternoon”. (Mar 2004)

- 5) Write the exact output of the following codes specifying RGB colours and output specifications in brackets. (Oct2004)

```
<html>
<had>
<title> computer shop </title>
</head>
<body>
<h3> Title : XYZ Computers Ltd. </h3>
<p> Address : Shakti Complex, Aurangabad </p>
<h1> <i> Dealers in: all types of peripherals </i>
</h1>
</body> </html>
```

- 6) Write HTML code for displaying a web page containing a six celled table as shown below (Oct2004)

Sachin	Sachin	Sachin
First	Second	Third

- 7) Write the exact output of the following HTML code with font specifications in brackets : (Mar2005)

```
<html>
<body>
<h1 align=”center”>LIST OF TOPICS </h1>
<hr>

 <p align = “left” > Operating System </p>
 <p align = “left” >Data Structure </p>
 <p align = “left” >C++ Programming </p>
 <p align = “left” > HTML </p>

<hr noshade>

 <p align =”center”> COMPUTER SCIENCE
</p>

<hr>
</body>
</html>
```

- 8) Write HTML code for the following (Mar 2005)

COMPUTER SCIENCE	PAPER -I	PAPER-II	TOTAL
	100	100	200

- 9) Write an exact output of the following HTML code with font specifications in brackets (Oct2005)

```
<html>
<h1> <center> HSC SYLLABUS </center> </h1>
<h3> PAPER I </h3>
To know more about
<p> Just click on here
<address>
 Data Structure
 GUI
 HTML
 C++
 VB
</address>
</body>
</html>
```

10) Write HTML code for web page displaying the following table : (Oct2005)

**Weather Forecast**

City	Wind speed	From	To
Mumbai	39	South	West
Raigad	47	South	West
Panaji	42	East	South

Speed in Km/hr.

11) Write the HTML code for the following: (Mar2006)

		Faculty		
		Arts	Science	Commerce
Students	Boys	100	400	500
	Girls	300	300	400

12) Write the exact output of the following HTML code with font specification in brackets: (Mar 2006)

```
<html>
<title>Introduction</title>
<body>
<h1> Computer Science </h1>
<hr> <u> E Balguru Samy </u>
<hr> <h4> Achyut S Godbole </h4>
</body>
</html>
```

13) Write a HTML code for the following: Government of Maharashtra <--Text size in h2

Education	
PWD	ADMN

for details

feel free to approach us

PWD is a link available where checking on PWD a webpage file 'Mahapwd.html' should be invoked. On clicking ADMN a webpage file 'admn.html' should be invoked. On clicking for details the user can write e-mail to mail-address 'free\_free@hscboard.com'. On clicking feel free to approach us, user can invoke to website 'www.maharashtra.gov.in'. (Oct2006)

14) Write the exact output of the following HTML code with font specification in brackets

```
<html>
<title> Examination </title>
<body>
<h1> First Semester Exam </h1>
<hr>
<u>MATHEMATICS</u>
<hr>
<u>STATISTICS</u>
<hr>
<u>BIOLOGY</u>
<hr>
<h5>MSEC BOARD </h5>
</body> </html>
```

15) Write the exact output of the following HTML code with font specifications in brackets : (Mar2007)

```
<html>
<body>
<h1> LIST OF BOOKS </h1> <hr>
<ul type="circle">
C++ Programming
HTML in Easy Steps
How to solve it By computer

<ol type="A">
 Microprocessor Programming
 Networking Essentials
 Microcontrollers

</body> </html>
```

16) Write the HTML code for the following table (Mar 2007)

SCIENCE		
F..Y.B.Sc.	S.Y.B.Sc.	T.Y.B.Sc.
800	600	200
ARTS		
F..Y.B.A.	S.Y.B.A.	T.Y.B.A.
500	400	200
COMMERCE		
F..Y.Bcom	S.Y.B.Com	T.Y.B.Com.
200	150	100

17) Write HTML code for the following (Oct2007) Subjects

- English (compulsory)
- Second Languages
  - Physics
  - Chemistry
    - Maths
    - Biology

18) Write the exact output of following HTML code

```
<html>
<head>
<title> </title>
</head>
<body>
<table border="2" width="50%">
<tr>
<td width="100%" colspan="2">
<h1 align="center"> HSC Board Exams</h1>
</td>
<tr>
<td width="50%" align="center">
<u>Paper I </u> </td>
<td width="50%" align="center">
<u> Paper II </u> </td> </tr>
<tr>
<td width="50%">
<p align="center">
<i>50 Marks </i> </td>
```

```
<td width="50%" align="center">
<i>50 Marks </i> </td>
</tr>
</table>
</body>
</html> (Oct2007)
```

19) Write the exact output of the following HTML code with font specifications in brackets:

```
<html>
<title> INTRODUCTION</title>
<body>
<h1> PCMBEC </h1>
<hr>
<u> NOBAL'S SERIES </u>
<h5>WELLKNOWN </h5>
</body>
</html> (Mar 2008)
```

20) Write HTML code for the following: (Mar2008)

		Year		
		2000	2001	2002
Sales	Units	500	1000	1500
	Income	Rs. 5000	Rs. 10000	Rs. 15000

21) Write the exact output of the following HTML code ( Oct 2008 )

```
<html>
<head>
<title> </title>
</head>
<body>
<h1> Languages </h1>

 English
 Second Languages

 Marathi
 Hindi
 French
 Sanskrit

</body>
</html>
```

22) Write HTML code for the following (Oct2008)

HSC Board Exams	
Computer	1. Paper I 2. Paper II
• 200 Marks	

23) Write HTML code for displaying following table on the web page : ( Mar 2009 )

S.N.	Student Name	Marks Obtained			Total
		Test-1	Test-2	Test-3	
1	ABC	150	150	125	425
2	LMN	129	130	131	390
3	XYZ	115	125	120	360

24) Write HTML code for the following output (Mar2009)

- ART
  - MARATHI
  - HINDI
  - ENGLISH
- COMMERCE
  - ACCOUNT
  - COSTING
  - AUDITING
- SCIENCE
  - PHYSICS
  - CHEMISTRY
  - MATHS
  - COMPUTER SC.

25) Write the HTML code for the following table (Oct 2009)

	Yuvraj 138*
Ind 387	Sehwag 83
	Peterson 58
Eng 238	Bopara 49
IND wins Ist ODI by 149	

26) Write the HTML code for the following table (Oct2009)

Year	Students		
	Boys	Girls	Total
2006	55	75	130
2007	75	95	170
Record			

27) Write the exact output of the following HTML code with font specifications in brackets (Mar2010)

```
<html>
<body>
<h1> <u> Network Connectivity Devices </u> </h1>

 Modem
 Hub
 Repeater
 Router

</body>
</html>
```

28) Write HTML code for the following: (Mar 2010)

		No. Books Purchased	
		F.Y.J.C.	S.Y.J.C.
Year	2004	1200	1300
	2005	1250	1400

29) Write exact output of the following HTML code with font specifications in brackets (Oct2010)

```
<html>
<body>
<h1>Terms used in Networking </h1>
<hr>
<ul type="circle">
 Bandwidth
 Attenuation
 Electromagnetic Interference

<ol type="a">
 Topology
 Ethernet
 Protocol

</body>
</html>
```

30) Write HTML code for the following :  
XII RESULT <—Text size h1

STREAM		
SCIENCE	COMMERCE	ART

SCIENCE is a link available where clicking on SCIENCE a web page file “SCIENCE.html” should be invoked, on clicking COMMERCE a web page file “COMMERCE.html” should be invoked and on clicking ART a web page file “ART.html” should be invoked. (Oct 2010)

31) Write a HTML code for the following output : (Mar2011)

COMPUTER DEVICES <— Text size h2  
align center

- INPUT DEVICES
  1. Keyboard
  2. Mouse
- STORAGE DEVICES
  1. Hard Disk
  2. Floppy Disk
  3. Compact Disk
- OUTPUT DEVICES
  1. Screen
  2. Printer

32) Write a HTML code for following (Mar2011)

COMPUTER SCIENCE	PAPER -I	PAPER-II	TOTAL
	100	100	200

33) Write the HTML code for the following (Oct2011)

V.I.P. SALES		
STATIONARY	ERASER	5
	PENCIL	7
	PEN	10
	BOOK	22
TOTAL		44

34) Write HTML code to display the following nested list : (Oct 2011)

- College
  - I. Arts
    - a. History
    - b. Politics
    - c. Languages
  - II. Science
    - a. Physics
    - b. Chemistry
    - c. Biology
  - III. Commerce
    - Accounts

- i. English
- ii. Marathi

35) Write exact output of the following HTML code with font specifications in bracket ((Mar2012)

```
<html>
<head>
<title> Computer Step </title>
</head>
<body>
<h2> <u> Title : Megastar Company Ltd. </u> </h2>
<p> Address : Surya Complex, Delhi </p>
<h4> Deals In : </h4>

 Software
 Hardware
 Peripherals

</body>
</html>
```

36) Write a HTML code for the following (Mar2012)

Roll No	Name	Marks		Total
		CS1	CS2	
1	RINA	75	70	145
2	RONAK	80	90	170
3	TINA	70	85	155

37) Write HTML code for a web page displaying of the following table : (Oct2012)

		Year		
		2007	2008	2009
Sales	Units	1500	400	1200
	Income	1000	900	3000

38) Write a HTML code to display the following output (Oct 2012)

List of Books <— Text size h1, default font is used

- How to solve it By computer
- HTML in Easy steps
- C++ Programming
  - A. Microprocessor Programming
  - B. Networking Essentials
  - C. Microcontrollers

39) Write exact output of the following HTML code with font specifications in brackets (Mar2013)

```
<html>
<body>
H1 align="center"> H.S.C. BOARD EXAM </h1>
<p align=center">
<i> SUBJECT : COMPUTER SCIENCE <i> </P>
<p align=center">
 THEORY AND PRACTICAL EXAMS
 </p>
<p align=center">
<u> TOTAL </u> ____200 MARKS </P>
</body>
</html>
```

Q) Which are different tags in HTML they do not have contains ending tags?

The HTML tag do not have contains ending tags as like follows:

- 1) < li >
- 2) < br >
- 3) < small >
- 4) < big >
- 5) < h1 > , < h2 > , ----- , < h6 >

40) Write HTML code for following: (Mar2013)

HTML COLOUR CODE IN HEXA-DECIMAL	
BLACK	WHITE
#000000	#FFFFFF