Lesson Number: 1 ***Writer: Vishal Verma***

Information Concept and Processing System Vetter:

1.1 OBJECTIVE

The objective of this lesson is to make the students familiar with the basics of Information and Data Processing System. After studying this lesson they will be familiar with:

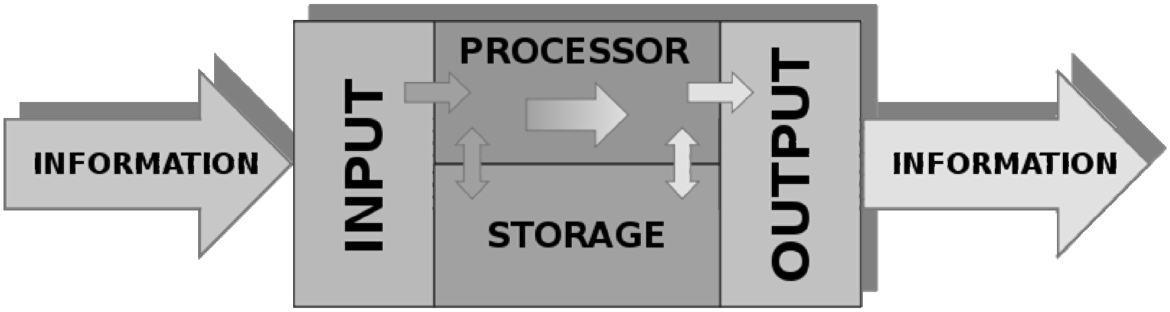
1. What is Information & Data?
2. What are the processing systems for data and information?
3. What is information processing language?

1.2 INTRODUCTION

An information processor or information processing system, as its name suggests, is a system (be it electrical, mechanical or biological) which takes information (a sequence of enumerated states) in one form and processes (transforms) it into another form, e.g. to statistics, by an algorithmic process.

An information processing system is made up of four basic parts, or sub-systems:

- input
- processor
- storage
- output



Anything in this world by which we can observe the change in one form of information into other, can be treated as an information system. Any object which in its one form is providing us the information in one form and after completion of some operation on same object the resultant object give us information in another form then the object in itself can be treated as an information system. Information science focuses on understanding problems from the perspective of the stakeholders involved and then applying information and other technologies as needed. In other words, it tackles systemic problems first rather than individual pieces of technology within that system. In this respect, information science can be seen as a response to technological determinism, the belief that technology "develops by its own laws, that it realizes its own potential, limited only by the material resources available, and must therefore be regarded as an autonomous system controlling and ultimately permeating all other subsystems of society." Within information science, attention has been given in recent years to human–computer interaction, groupware, the semantic web, value sensitive design, iterative design processes and to the ways people generate, use and find information. Information science should not be confused with information theory, the study of a particular mathematical concept of information

1.3 EVOLUTION OF INFORAMTION PROCESSING

Information processing is the change (processing) of information in any manner detectable by an observer. As such, it is a process which *describes* everything which happens (changes) in the universe, from the falling of a rock (a change in position) to the printing of a text file from a digital computer system. In the latter case, an information processor is changing the form of presentation of that text file. Information processing may more specifically be defined in terms used by Claude E. Shannon as the conversion of latent information into manifest information. Latent and manifest information is defined through the terms of equivocation (remaining uncertainty, what value the sender has actually chosen), dissipation (uncertainty of the sender what the receiver has actually received) and transformation (saved effort of questioning - equivocation minus dissipation).

Within the field of cognitive psychology, information processing is an approach to the goal of understanding human thinking. It arose in the 1940s and 1950s. The essence of the approach is to see cognition as being essentially computational in nature, with *mind* being the *software* and the brain being the *hardware*. The information processing approach in psychology is closely allied to cognitive in psychology and functionalism in philosophy although the terms are not quite synonymous. Information processing may be sequential or parallel, either of which may be centralized or decentralized (distributed). The parallel distributed processing approach of the mid-1980s became popular under the name connectionism. In the early 1950s Friedrich Hayek was ahead of his time when he posited the idea of spontaneous order in the brain arising out of *decentralized networks of simple units* (neurons). However, Hayek is rarely cited in the literature of connectionism.

In the 1970s, Abraham Moles and Frieder Nake were among the first to establish and analyze links between information processing and aesthetics. While discussing the topic of information processing, the discussion about the information science and theory also play its important role since it helps us to understand technically about the processing strategies of system. Information science is an interdisciplinary science primarily concerned with the analysis, collection, classification, manipulation, storage, retrieval and dissemination of information. Practitioners within the field study the application and usage of knowledge in organizations, along with the interaction between people, organizations and any existing information systems, with the aim of creating, replacing, improving or understanding information systems. Information science is often (mistakenly) considered a branch of computer science. However, it is actually a broad, interdisciplinary field, incorporating not only aspects of computer science, but often diverse fields such as archival science, cognitive science, commerce, communications, law, library science, musicology, management, mathematics, philosophy, public policy, and the social sciences.

The information processing theory approach to the study of cognitive development evolved out of the American experimental tradition in psychology.

Information processing theorists proposed that like the computer, the human mind is a system that processes information through the application of logical rules and strategies. Like the computer, the mind has a limited capacity for the amount and nature of the information it can process.

Finally, just as the computer can be made into a better information processor by changes in its hardware (e.g., circuit boards and microchips) and its software (programming), so do children become more sophisticated thinkers through changes in their brains and sensory systems (hardware) and in the rules and strategies (software) that they learn.

Beliefs of the information-processing approach

1. When the individual perceives, encodes, represents, and stores information from the environment in his mind or retrieves that information, he or she is thinking. Thinking also includes responding to any constraints or limitations on memory processes.

2. The proper focus of study is the role of change mechanism in development. Four critical mechanisms work together to bring about change in children's cognitive skills: encoding, strategy construction, automatization, and generalization. To solve problems effectively, children must encode critical information about a problem and then use this encoded information and relevant prior knowledge to construct a strategy to deal with the problem

3. Development is driven by self-modification. Like Piaget's theory of cognitive development, the information-processing approach holds that children play an active role in their own development. Through self-modification, the child uses knowledge and strategies she has acquired from earlier problem solution to modify her responses to a new situation or problem. In this way, she builds newer and more sophisticated responses from prior knowledge

4. Investigators must perform careful task analysis of the problem situations they present to children. According to this view, not only the child's own level of development but the nature of the task itself constraints child's performance. Thus a child may possess the basic ability necessary to perform a particular task when it is presented in a simple form, without unnecessary complexities. However, if extra or misleading information is added to the same task, the child may become confused and be unable to perform it.

Structure of the information-processing system

In the store model of the human information-processing system, information from the environment that we acquire through our senses enter the system through the sensory register.

- The store model: A model of information processing in which information is depicted as moving through a series of processing units — sensory register, short-term memory, long-term memory — in each of which it may be stored, either fleetingly or permanently.

- Sensory register: the mental processing unit that receives information from the environment and stores it fleetingly.

- Short-term memory: the mental processing unit in which information may be stored temporarily; the work space of the mind, where a decision must be made to discard information or to transfer it to permanent storage, in long-term memory.

- Long-term memory**:** the encyclopedic mental processing unit in which information may be stored permanently and from which it may be later retrieved.

1.4 DATA PROCESSING

Computer data processing is any process that uses a computer program to enter data and summarize, analyze or otherwise convert data into usable information. The process may be automated and run on a computer. It involves recording, analyzing, sorting, summarizing, calculating, disseminating and storing data. Because data is most useful when well-presented and actually *informative*, data-processing systems are often referred to as information systems. Nevertheless, the terms are roughly synonymous, performing similar conversions; data-processing systems typically manipulate raw data into information, and likewise information systems typically take raw data as input to produce information as output.

Data processing may or may not be distinguished from data conversion, when the process is merely to convert data to another format, and does not involve any data manipulation.

Data analysis

When the domain from which the data are used is a science or an engineering field, data processing and information systems are considered terms that are too broad and the more specialized term data analysis is typically used. This is a focus on the highly-specialized and highly-accurate algorithmic derivations and statistical calculations that are less often observed in the typical general business environment. In these contexts data analysis packages like DAP, gretl or PSPP are often used. This divergence of culture is exhibited in the typical numerical representations used in data processing versus numerical; data processing's measurements are typically represented by integers or by fixed-point or binary-coded decimal representations of numbers whereas the majority of data analysis's measurements are often represented by floating-point representation of rational numbers.

Processing

Basically, data is nothing but unorganized facts and which can be converted into useful information. This process of converting facts to information is Processing.

Practically all naturally occurring processes can be viewed as examples of data processing systems where "observable" information in the form of pressure, light, etc. are converted by human observers into electrical signals in the nervous system as the senses we recognize as touch, sound, and vision. Even the interaction of non-living systems may be viewed in this way as rudimentary information processing systems. Conventional usage of the terms *data processing* and *information system*s restricts their use to refer to the algorithmic derivations, logical deductions, and statistical calculations that recur perennially in general business environments, rather than in the more expansive sense of all conversions of real-world measurements into real-world information in, say, an organic biological system or even a scientific or engineering system. In order to be processed by a computer, data needs first be converted into a machine readable format. Once data is in digital format, various procedures can be applied on the data to get useful information. Data processing may involve various processes, including:

- Data summarization
- Data aggregation
- Data validation
- Data tabulation
- Statistical analysis

1.5 INFORMATION LANGUAGE AND COMMUNICATION

Information Processing Language (IPL) is a programming language developed by Allen Newell, Cliff Shaw, and Herbert Simon at RAND Corporation and the Carnegie Institute of Technology from about 1956. Newell had the role of language specifier-application programmer, Shaw was the system programmer and Simon took the role of application programmer-user.

The language includes features intended to support programs that could perform general problem solving, including lists, associations, schemas (frames), dynamic memory allocation, data types, recursion, associative retrieval, functions as arguments, generators (streams), and cooperative multitasking. IPL pioneered the concept of list processing, albeit in an assembly-language style.

Computer that uses IPL has some common features as:

1. a set of *symbols*. All symbols are addresses, and name cells. Unlike symbols in later languages, symbols consist of a character followed by a number, and are written H1, A29, 9-7, 9-100.
    1. Cell names beginning with a letter are *regional*, and are absolute addresses.
    2. Cell names beginning with "9-" are *local*, and are meaningful within the context of a single list. One list's 9-1 is independent of another list's 9-1.
    3. Other symbols (e.g., pure numbers) are *internal*.
2. a set of *cells*. Lists are built from several cells holding mutual references. Cells have several fields:
    1. P, a 3-bit field used for an operation code when the cell is used as an instruction, and unused when the cell is data.
    2. Q, a 3-valued field used for indirect reference when the cell is used as an instruction, and unused when the cell is data.
    3. SYMB, a symbol used as the value in the cell.
3. a set of *primitive processes*, which would be termed *primitive functions* in modern languages.

The main data structure of IPL is the list, but lists are more intricate structures than in many languages. A list consists of a singly-linked sequence of symbols, as might be expected -- plus some *description lists*, which are subsidiary singly-linked lists interpreted as alternating attribute names and values. IPL provides primitives to access and mutate attribute value by name. The description lists are given local names (of the form 9-1). So, a list called L1 holding the symbols S4 and S5, and described by associating value V1 to attribute A1 and V2 to A2, would be stored as follows. 0 indicates the end of a list; the cell names 100, 101, etc. are automatically generated internal symbols whose values are irrelevant. These cells can be scattered throughout memory; only L1, which uses a regional name that must be globally known, needs to reside in a specific place.

IPL has a library of some 150 basic operations. These include such operations as:

- Test symbols for equality
- Find, set, or erase an attribute of a list
- locate the next symbol in a list; insert a symbol in a list; erase or copy an entire list.
- Arithmetic operations (on symbol names).
- Manipulation of symbols; e.g., test if a symbol denote an integer, or make a symbol local.
- I/O operations
- "generators", which correspond to iterations and filters in functional programming. For example, a generator may accept a list of numbers and produce the list of their squares. Generators could accept suitably-designed functions -- strictly, the addresses of code of suitably-designed functions -- as arguments.

Lesson Number: 2 ***Writer: Vishal Verma***

Elements of Computer Processing System ***Vetter:***

2.1 OBJECTIVE

The objective of this lesson is to make the students familiar with the basics of Computer Processing System. After studying this lesson they will be familiar with:

1. What is Computer & its Units?
2. What are Hardware & Software Components of Computers?
3. What is Software and its type?

2.2 INTRODUCTION

A **computer** is a programmable machine that receives input, stores and manipulates data, and provides output to the user. The word computer is a combination of hardware components and software. In alone neither hardware nor software can’t do anything for the user.

To be familiar with the computer we start our discussion form the hardware components of the computer and latter on we do discuss the software components. Anatomy of computer components can be understood by the diagram as shown in Fig.1 below:

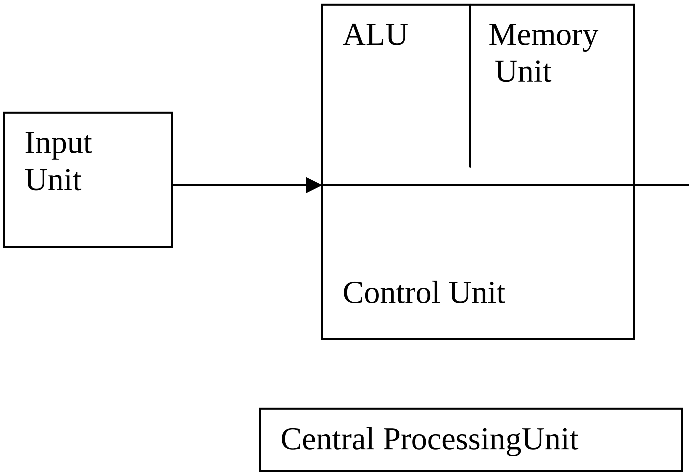


Fig. 1

2.3 HARDWARE

As we have already discussed that computer is the combination of hardware and software so now we will discuss various hardware components of the computers.

2.3.1 CPU

CPU stands for the Central Processing Unit. This unit is responsible controlling almost all of the important tasks of the computer. It is also known as the brain of the computer. As shown in the Fig.1 CPU of the computer is made-up of the basic three components as

i) ALU : ALU stands for Arithmetic Logic Unit . It take care of all the arithmetic and logical operation of the computer. This unit provide the basic calculation power to the CPU. All requests for the calculation from the user as well as from the other components of computer is managed by the ALU.

ii) Memory Unit : This unit provide the work space to the computer, to understand the importance of memory unit you can imagine the situation of opening the multiple paper files by a user on the table, if the size of the table is large than it become easy for the user to work with more than one files simultaneously, various file operations like opening file, reading from file, writing into file etc can be done easily if more and more memory space is available. It must be kept be mind that computer need memory to perform any operation with any file, importantly even the operating systems file's are also opened and executed first into the memory and then they control the working or rest of the components of the computer system

iii) Control Unit: This unit is responsible for controlling the working and making coordination of other parts and units of the computer system. The term processor is generally used for the control unit of control unit of the computer. The speed of the control unit is measured in terms of number of instructions executed per unit of time.

2.3.2 STORAGE DEVICES AND MEDIA

The storage capacity of the main memory is very limited. Often it is necessary to store hundreds of millions of bytes of data for the CPU to process. Therefore additional memory is required in all the computer systems. This memory is called *auxiliary memory* or *secondary storage*.

In this type of memory the cost per bit of storage is low. However, the operating speed is slower than that of the primary storage. Huge volume of data are stored here on permanent basis and transferred to the primary storage as and when required. Most widely used secondary storage devices are *magnetic tapes* and *magnetic disk*.

2.3.2.1 MAGANETIC TAPE

Magnetic tapes are used for large computers like mainframe computers where large volume of data is stored for a longer time. In PC also you can use tapes in the form of cassettes. The cost of storing data in tapes is inexpensive. Tapes consist of magnetic materials that store data permanently. It can be 12.5 mm to 25 mm wide plastic film-type and 500 meter to 1200 meter long which is coated with magnetic material. One important point must be remembered abtout the basic construction of the magnetic disk is that the whole space is divided into small blocks of small and large size : Small size blocks are knows as inter block gap(IBG) and these are used to control the speed of the tape when is read and write onto the tape, and the blocks of large size are used to store the physical date on to the tape. In Fig 2 the dark blocks are representing the IBG and white blocks are representing the blocks used for data storage. The deck is connected to the central processor and information is fed into or read from the tape through the processor. It similar to cassette tape recorder.

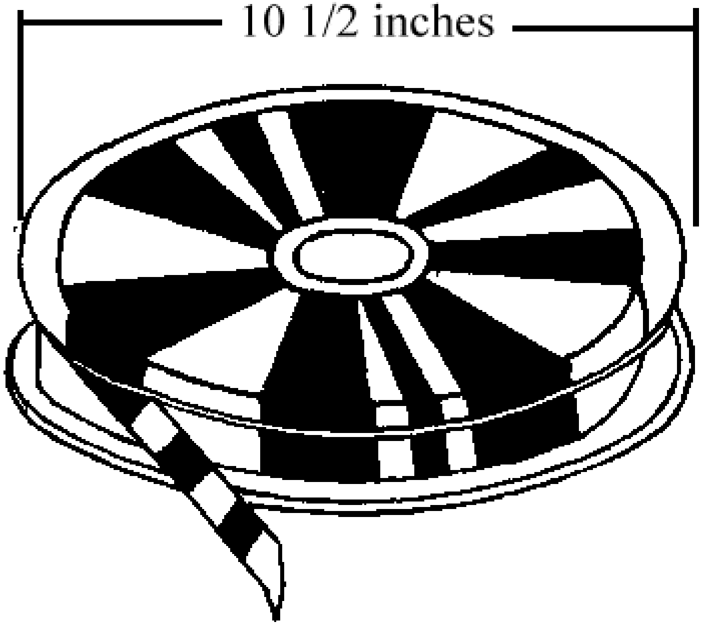


Fig. 2

Advantages of Magnetic Tape:

- **Compact:** A 10-inch diameter reel of tape is 2400 feet long and is able to hold 800, 1600 or 6250 characters in each inch of its length. The maximum capacity of such tape is 180 million characters. Thus data are stored much more compactly on tape.
- **Economical**: The cost of storing characters is very less as compared to other storage devices.
- **Fast**: Copying of data is easier and fast.
- **Long term Storage and Re-usability**: Magnetic tapes can be used for long term storage and a tape can be used repeatedly with out loss of data.

2.3.2.2 MAGANETIC DISK

You might have seen the gramophone record, which is circular like a disk and coated with magnetic material. Magnetic disks used in computer are made on the same principle. It rotates with very high speed inside the computer drive. Data is stored on both the surface of the disk. Magnetic disks are most popular for *direct access* storage device*.* Each disk consists of a number of invisible *concentric circles* called *tracks*. Information is recorded on tracks of a disk surface in the form of tiny magnetic spots. The presence of a magnetic spot represents *one bit* and its absence represents zero bit. The information stored in a disk can be read many times without affecting the stored data. So the reading operation is non-destructive. But if you want to write a new data, then the existing data is erased from the disk and new data is recorded.

2.3.2.2.1 FLOPPY DISK

It is similar to magnetic disk discussed above. They are 5.25 inch or 3.5 inch in diameter. They come in single or double density and recorded on one or both surface of the diskette. The capacity of a 5.25-inch floppy is 1.2 mega bytes whereas for 3.5 inch floppy it is 1.44 mega bytes. It is cheaper than any other storage devices and is portable. The floppy is a low cost device particularly suitable for personal computer system.

Fig 3

2.3.2.3 OPTICAL DISK

With every new application and software there is greater demand for memory capacity. It is the necessity to store large volume of data that has led to the development of optical disk storage medium. Optical disks can be divided into the following categories:

1. *Compact Disk/ Read Only Memory (CD-ROM)*: CD-ROM disks are made of reflective metals. CD-ROM is written during the process of manufacturing by high power *laser beam.* Here the storage density is very high, storage cost is very low and access time is relatively fast. Each disk is approximately 4 1/2 inches in diameter and can hold over

600 MB of data. As the CD-ROM can be *read only* we cannot write or make changes into the data contained in it.

2. *Write Once, Read Many (WORM)*: The inconvenience that we can not write any thing in to a CD-ROM is avoided in WORM. A WORM allows the user to write data permanently on to the disk. Once the data is written it can never be erased without physically damaging the disk. Here data can be recorded from keyboard, video scanner, OCR equipment and other devices. The advantage of WORM is that it can store vast amount of data amounting to gigabytes (10^9 bytes). Any document in a WORM can be accessed very fast, say less than 30 seconds.
3. *Erasable Optical Disk*: These are optical disks where data can be written, erased and re-written. This also applies a laser beam to write and re-write the data. These disks may be used as alternatives to traditional disks. Erasable optical disks are based on a technology known as *magnetic optical* (MO). To write a data bit on to the erasable optical disk the MO drive's laser beam heats a tiny, precisely defined point on the disk's surface and magnetizes it.

2.4 VDU

The most popular output device is the Visual Display Unit (VDU). It is also called the monitor. A Keyboard is used to input data and Monitor is used to display the input data and to receive messages from the computer. It is mainly used to represent the data in the pictorial form to the user. A monitor has its own box which is separated from the main computer system and is connected to the computer by cable. In some systems it is compact with the system unit. It can be *color* or *monochrome*. In the early time we have only monochrome CRT monitors with us but with the improvement in technology now a days we have colored monitors of various types like CRT,LCD, Plasma Panel etc with us to have a wider and more clear images. VDU which can display even 3 D images are also available in the market.

2.5 INPUT-OUTPUT DEVICES

A computer is only useful when it is able to communicate with the external environment. When you work with the computer you feed your data and instructions through some devices to the computer. These devices are called Input devices. Similarly computer after processing it gives output through other devices called output devices.

For a particular application one form of device is more desirable compared to others. We will discuss various types of I/O devices that are used for different types of applications. They are also known as peripheral devices because they surround the CPU and make a communication between computer and the outer world.

2.5.1 INPUT DEVICES

Input devices are necessary to convert our information or data into a form which can be understood by the computer. A good input device should provide timely, accurate and useful data to the main memory of the computer for processing followings are the most useful input devices.

1. **Keyboard: -** This is the standard input device attached to all computers. The layout of keyboard is just like the traditional typewriter of the type QWERTY. It also contains some extra command keys and function keys. It contains a total of 101 to 104 keys. A typical keyboard used in a computer is shown in Fig. 4 You have to press correct combination of keys to input data. The computer can recognize the electrical signals corresponding to the correct key combination and processing is done accordingly.

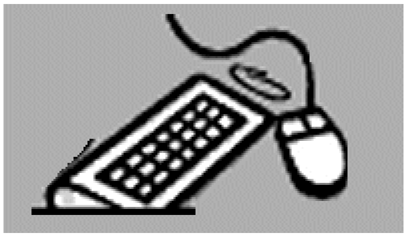


Fig. 4

2. **Mouse:** - Mouse is an input device shown in Fig. 5 that is used with your personal computer. It rolls on a small ball and has two or three buttons on the top. When you roll the mouse across a flat surface the screen censors the mouse in the direction of mouse movement. The cursor moves very fast with mouse giving you more freedom to work in any direction. It is easier and faster to move through a mouse.

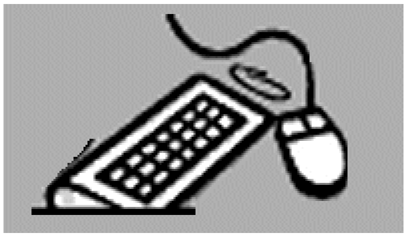


Fig. 5

3. **Scanner:** The keyboard can input only text through keys provided in it. If we want to input a picture the keyboard cannot do that. Scanner is an optical device that can input any graphical matter and display it back. The common optical scanner devices are Magnetic Ink Character Recognition (MICR), Optical Mark Reader (OMR) and Optical Character Reader (OCR).

   - **Magnetic Ink Character Recognition (MICR):** - This is widely used by banks to process large volumes of cheques and drafts. Cheques are put inside the MICR. As they enter the reading unit the cheques pass through the magnetic field which causes the read head to recognise the character of the cheques.

- **Optical Mark Reader (OMR):** This technique is used when students have appeared in objective type tests and they had to mark their answer by darkening a square or circular space by pencil. These answer sheets are directly fed to a computer for grading where OMR is used.
- **Optical Character Recognition (OCR):** - This technique unites the direct reading of any printed character. Suppose you have a set of hand written characters on a piece of paper. You put it inside the scanner of the computer. This pattern is compared with a site of patterns stored inside the computer. Whichever pattern is matched is called a character read. Patterns that cannot be identified are rejected. OCRs are expensive though better the MICR.

2.5.2 OUTPUT DEVICES

1. **Visual Display Unit:** The most popular input/output device is the Visual Display Unit (VDU). It is also called the monitor. A Keyboard is used to input data and Monitor is used to display the input data and to receive messages from the computer. A monitor has its own box which is separated from the main computer system and is connected to the computer by cable. In some systems it is compact with the system unit. It can be *color* or *monochrome*.
2. **Terminals:** It is a very popular interactive input-output unit. It can be divided into two types: hard copy terminals and *soft copy* terminals. A *hard copy* terminal provides a printout on paper whereas soft copy terminals provide visual copy on monitor. A terminal when connected to a CPU sends instructions directly to the computer. Terminals are also classified as dumb terminals or intelligent terminals depending upon the work situation.
3. **Printer:** It is an important output device which can be used to get a printed copy of the processed text or result on paper. There are different types of printers that are designed for different types of applications. Depending on

their speed and approach of printing, printers are classified as *impact* and *non-impact* printers. Impact printers use the familiar typewriter approach of hammering a typeface against the paper and inked ribbon. *Dot-matrix printers* are of this type. Non-impact printers do not hit or impact a ribbon to print. They use electro-static chemicals and ink-jet technologies. *Laser printers* and *Ink-jet printers* are of this type. This type of printers can produce color printing and elaborate graphics.

2.6 DATA COMMUNICAITON EQUIPMENTS

Data communication equipments are the equipments which are used in/with computer system to transfer the data from one location to another location. In this era of networking the transfer of data from one machine to another (at any distant location) is possible because of availability of communication devices like:

Network Cards : These are first and basic components of the computer which are used for identification of a single machine in the group of machines and ultimately provide the means of data communication.

Switch/Hub : These components are used for connecting the two or more system together at the work place. They play their important role when you have to establish the LAN networking in the office or lab etc. The primary difference between their working is that a switch make it possible to use the same data transfer speed available at all the systems in the network while the hub distribute the incoming speed of data transfer at the server equally to the system connected together in that LAN.

Router : This is very important equipment in the data communication system as this is used to make a match between two networks which are working or operation on two different networks (networks using two different type of protocols of transfer of data) .

2.7 SOFTWARE

As you know computer cannot do anything without instructions from the user. In order to do any specific job you have to give a sequence of instructions to the computer. This set of instructions is called a computer *program*. Software refers to the set of computer programs, procedures that describe the programs, how they are to be used. We can say that it is the collection of programs, which increase the capabilities of the hardware. Software guides the computer at every step where to start and stop during a particular job. The process of software development is called *programmin*g.

You should keep in mind that software and hardware are complementary to each other. Both have to work together to produce meaningful result. Another important point you should know that producing software is difficult and expensive.

SOFTWARE TYPES

Computer software is normally classified into two broad categories.

- Application Software
- System software

2.7.1 APPLICATION SOFTWARE

Application Software is a set of programs to carry out operations for a specific application. For example, payroll is application software for an organization to produce pay slips as an output. Application software is useful for word processing, billing system, accounting, producing statistical report, analysis of numerous data in research, weather forecasting, etc. In later modules you will learn about MS WORD, Lotus 1-2-3 and dBASE III Plus. All these are application softwares.

Another example of application software is programming language. Among the programming languages COBOL (Common Business Oriented Language) is more suitable for business application whereas FORTRAN (Formula Translation) is useful for scientific application. We will discuss about languages in next section.

2.7.2 SYSTEM SOFTWARE

You know that an instruction is a set of programs that has to be fed to the computer for operation of computer system as a whole. When you switch on the computer the programs written in ROM is executed which activates different units of your computer and makes it ready for you to work on it. This set of program can be called system software. Therefore system software may be defined as a set of one or more programs designed to control the operation of computer system.

System software are general programs designed for performing tasks such as controlling all operations required to move data into and out of the computer. It communicates with printers, card reader, disk, tapes etc. monitor the use of various hardware like memory, CPU etc. Also system software are essential for the development of applications software. System Software allows application packages to be run on the computer with less time and effort. *Remember that it is not possible to run application software without system software*.

Development of system software is a complex task and it requires extensive knowledge of computer technology. Due to its complexity it is not developed in house. Computer manufactures build and supply this system software with the computer system. DOS, UNIX and WINDOWS are some of the widely used system software. Out of these UNIX is a multi-user operating system whereas DOS and WINDOWS are PC-based. We will discuss in detail about DOS and WINDOWS in the next module.

So without system software it is impossible to operate your computer. The following picture is shown in Fig. 3.1 relation between hardware, software and you as a user of computer system.

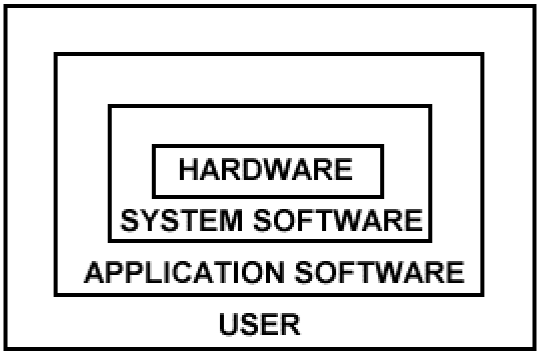


Fig. 6   Relation between hardware, software.

Lesson Number: 4 ***Writer: Vishal Verma***

Operating System ***Vetter:***

4.1 OBJECTIVE

The objective of this lesson is to make the students familiar with the basics of Computer's Operating System. After studying this lesson they will be familiar with:

1. Role of O.S as a resource manager?
2. Concept of Priority, Projection and Parallelism?
3. Basics of DOS and UNIX?
4. GUI Window

4.2 INTRODUCTION

When the computer is powered on a software known as the operating system starts automatically and it takes the control of the machine. An Operating System is a set of programs that help in controlling and managing the Hardware and the Software resources of a computer system. A good operating system should have the following features;

1. Help in the loading of programs and data from external sources into the internal memory before they are executed.
2. Help programs to perform input/output operations, such as;
   - Print or display the result of a program on the printer or the screen.
   - Store the output data or programs written on the computer in storage device.
   - Communicate the message from the system to the user through the VDU.
   - Accept input from the user through the keyboard or mouse.

4.3 CONCEPT OF OPERATING SYSTEM

As the name suggests, the operating System is used for operating the system or the computer. It is a set of computer programs and also known as DOS (Disk Operating System). The main functions of DOS are to manage disk files, allocate system resources according to the requirement. DOS provides features essential to control hardware devices such as Keyboard, Screen, Disk Devices, Printers, Modems and programs.

Basically, O.S is the medium through which the user and external devices attached to the system communicate with the system. O.S translate the command issued by the user in the format that is understandable by the computer and instruct computer to work accordingly. It also translates the result and any error message in the format for the user to understand.

4.3.1 OPERATING SYSTEM AS RESOURCE MANAGER PROCESSOR

When multiple users or multiple jobs running concurrently, resources must be allocated to each of them Many types of resources - Some (such as CPU cycles ,main memory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code are available on the system, depending upon the requirement of the current process the required resources must be allocated to it. It is only the allocation of resources to the process which make it possible for the process to take part in the competition to get the processor for execution. There exist a number of algorithms to make a check and keep control the time period for which the resources remained allocated to the processes. The current position of the process (depending upon the allocated resources) in the ready queue or in the waiting queue is also decided by the resource manager of the operating system. This manager is supposed to keep track of all the resources of the system at all of the time. The addition of new resources and removal of existing resources is also done by the resource manager. All the processes whether new or currently executing are required to make request only and only to the resource manager for the required resources.

4.3.2 OPERATING SYSTEM AS DEVICE MANAGER

Devices are the important part of the computer system, because they are connected with the system hence are also known as peripherals, depending upon the available communication ports on the system we can attach only a limited number of devices with the system. The processes always raise some request for the devices to the O.S, hence the device manager part of the O.S ensures the assignment of the devices to the processes, during this allocation procedure it has to take care about the time period for which the devices are allocated and the order in which same device is allocated to number of processes so that dead lock like situation can be avoided or removed from the system.. The availability of multiple instances is very helpful for device manager of the system to fulfill the request of processes in simultaneous fashion. Device manager also uses some algorithm for make a good rotation for the assignment of device to processes, Round robin is one of the popular algorithm which is used for assignment of devices to processes. It also take care for the processes in the waiting queue of the system so that they should fall in the infinite wait time just because of waiting for a device

4.3.3 OPERATING SYSTEM AS MEMORY MANAGER

Current computer architectures arrange the computer's memory in a hierarchical manner, starting from the fastest registers, CPU cache, random access memory and disk storage(secondary storage). An operating system's memory manager coordinates the use of these various types of memory by tracking which one is available, which is to be allocated or de-allocated and how to move data between them. This activity, usually referred to as virtual memory management, increases the amount of memory available for each process by making the disk storage seem like main memory. There is a speed penalty associated with using disks or other slower storage as memory – if running processes require significantly more RAM than is available, the system may start thrashing. This can happen either because one process requires a large amount of RAM or because two or more processes compete for a larger amount of memory than is available. This then leads to constant transfer of each process's data to slower storage.

Another important part of memory management is managing virtual addresses. If multiple processes are in memory at once, they must be prevented from interfering with each other's memory (unless there is an explicit request to utilize shared memory). This is achieved by having separate address spaces. Each process sees the whole virtual address space, typically from address 0 up to the maximum size of virtual memory, as uniquely assigned to it. The operating system maintains a page-table that matches virtual addresses to physical addresses. These memory allocations are tracked so that when a process terminates, all memory used by that process can be made available for other processes. The operating system can also write inactive memory pages to secondary storage. This process is called "paging" or "swapping" – the terminology varies between operating systems.

It is also typical for operating systems to employ otherwise unused physical memory as a page cache; requests for data from a slower device can be retained in memory to improve performance. The operating system can also pre-load the in-memory cache with data that may be requested by the user in the near future; SuperFetch is an example of this.

4.4 PRIORITIES, PROJECTION AND PARALLELISM

These concepts are the features of operating system which helps to identify the efficiency and speed up factor of system. These features if available or supported by the system then user can achieve better output from the system. All of these features are discussed in detail here :

4.4.1 CONCEPT OF PRIORITY IN OPERATING SYSTEM

Term Priority can be used to describe the importance of a process among all the available ones. Operating systems have the scheduling methods (algorithms) to execute the processes as per pre-decided criteria e.g first come first serve(FCFS), shortest job first(SJF) etc, in these types of algorithms the user is bound by the system to get the output as per the sequence only, in case of an emergent requirement he/she doesn’t has any alternate to execute the process before its turn. To overcome this limitation faced by the users a new term priority is introduced by which user or system can assign a priority level to the process, and system is so designed that it is capable to execute the processes with high priority first than other processes in the queue. Fixed priority pre-emptive scheduling is one of the algorithm which works on the basis of priority of processes. Generally the priority is assigned in term of natural numbers starting from 0. In some system the 0 is considered as the highest priority number and in others the highest number in sequence is considered as highest priority level.

4.4.2 CONCEPT OF PROJECTION IN OPERATING SYSTEM

Term projection is used to specify image of uncompleted processes, when the system is executing the processes by using virtual memory or the paging to compensate the non availability of RAM then the situation which occurs most frequently is that; at the time of read/write operation of a process it is snatched from the RAM and is stored in the virtual memory, the data which helps to restart the process from its lastly executed statement is referred to as the projection of that process in the memory, it helps the system to keep track of the processes which have completed their execution and which are not completed.

4.4.3 CONCEPT OF PARALLELISM IN OPERATING SYSTEM

Parallelism is used to keep the system busy all of the time and is achieved by executing more than one process at the same time. When system execute the processes in parallel then the CPU will remain busy almost all of the time, during this if any of the process switch for input/output of data then system start executing another process from the queue. A number of algorithms are available to achieve the high level of parallelism on the systems. From the user point of view no doubt the machine is executing all the processes in parallel but in reality the CPU will entertain only one process at a unit of time, but because of small fraction of time unit and fast switching among the processes the user is not capable of identify which process is under execution at the moment. “Round robin” is one of the commonly used algorithm for the purpose. Parallelism is achieved at the many levels like bit level, instruction level, function level process level etc.

4.5 COMMAND INTERPRETER

A command interpreter is the part of a computer operating system that understands and executes commands that are entered interactively by a human being or from a program. In some operating systems, the command interpreter is called the shell. Command-line interpreters allow users to issue various commands in a very efficient (and often terse) way. This requires the user to know the names of the commands and their parameters, and the syntax of the language that is interpreted

4.6 TYPICAL COMMANDS OF DOS/UNIX/NET WARE

DIR COMMAND

The DIR command gives the list of directories that exist on the disk that is mounted on the active drive.

Syntax : C:\> DIR A:\> DIR

MAKING OR CREATING DIRECTORY

The MD or MKDIR command creates a new empty directory whose name is the last item specified in the pathname, in the specified drive. If active, the drive need not be specified. If the directory is to be created as a sub-directory of the working directory on the active drive, typing MD {directory name} at the DOS prompt or command prompt is sufficient.

Examples:

# 1. A:\> MD \ACCT\SALARY

makes a SALARY directory in the: drive, under ACCT directory.

# 2. A:\> MD C:\> SALARY

Makes a salary directory in the C: drive, under root directory.

DELETING A DIRECTORY

You may want to delete or remove a directory to simplify your directory structure. DOS provides RD (Remove Directory) to delete a directory.

Example:

# 1. A:\> RD \ACCT\SALARY

removes the SALARY sub-directory in ACCT directory.

NOTE: You cannot delete a directory if you are in it. Before you can delete a directory, you must type cd.. at the command prompt. At the same point of time, the directory to be deleted should be empty.

COPYING FILES

To copy a file, DOS provides `COPY' command. When you use `copy' command, you must use the following two parameters; the location and the name of the file you want to copy, or the source; and the location and the file name to which you want to copy the file or the target (destination). You separate the source and the destination or target with a space. The syntax of the `COPY' command is

```
COPY  {source}  {destination}  or,
COPY [drive:] [path] [filename]  [drive:] [path] [filename]
```

i.e. the first set of drive, path and filename refers to the source file, and the second set of drive, path and filename
refers to the destination file.

USE OF WILDCARD CHARACTERS

If you want to carry out a task for a group of files whose names have something in common, you can use wildcard characters to specify groups of files. DOS recognize two wildcard characters: asterisk (*) represents one or more characters that a group of files has in common; and the question mark (?) represents a single character that a group of files has in common. You can use wildcards to replace all or part of a file's name or its extension. The following table shows examples of wildcards:

Use of wildcard characters in COPY command

```
# 1. A:\>COPY \letters\*.COB B:
```

It means, copy all files with extension *.COB from the directory LETTERS under the ROOT directory to the working or ROOT directory of the `B' drive.

```
# 2. A:\> COPY B:\COMPANY\OPEL.*
```

The command is to copy all files with primary name OPEL (irrespective extension) in the directory COMPANY under ROOT of the drive `B' into the current working directory of the disk mounted in `A' drive. Incase of one drive, the system will ask for the source and
target drive.

The command,

#3.A:\>COPY C:\*.*
copies all files of the ROOT directory of the 'C' drive into the working directory of the 'A' drive.

RENAMING FILES

To rename a file, DOS provides REN command. The REN command stands for "Rename". When you use the REN command, you must include two parameters. The first is the file you want to rename, and the second is the new name for the file. You separate the two names with a space. The REN command follows this pattern:

REN oldname newname

Example: REN NOS.DOC NOS.MEM

Rename the old filename NOS.DOC to a new filename NOS.MEM.

DELETING FILES

This section explains how to delete or remove a file that is no longer required in the disk. DOS provides DEL command, which means to delete.

Syntax : DEL {drive:} {path} {filename}

Example:

# 1. DEL \DOS\EDIT.HLP

delete the EDIT.HLP from the DOS directory under ROOT directory.

4.7 GUI WINDOWS

Graphical User Interface: As compared DOS, Windows provides user-friendlier interface to work on. Its improved graphical user interface makes learning and using Windows more natural and easier for all types of users. It is more powerful, customizable and efficient.
GUI Window provides the following user friendly features as :

Start Button

Introduction of START button in Windows made life much simpler while there is a need to access multiple programs. It is the gateway of accessing most of the functionality available in the computer loaded with Windows. Just Click on the Start button anytime to start any programs, open or find documents, change windows settings, get Help, manage Files, maintain system, and much more.

Taskbar

As the name suggests, the Task bar provides information and access to the entire task that has been currently activated by Windows. Using this one can keep a track of what all programs have been activated and switched between them.

Windows Explorer

Windows Explorer is more or less acts as a File Manager for Windows, but with lots of new features. It is more efficient, faster and user friendly as compared to File Manager of DOS. Using Explorer one can easily browse through all the drives and network resources available and manage files.

Right Mouse Button

Clicking on the right mouse button activates a pop-up menu in any programs so as to help in completing a task efficiently.

Long File Names

As the Ms-DOS convention follows, non-of the file used in DOS environment should be more than 8 characters of primary name and optional secondary name (extension) of three characters. However Windows has broken this barrier. Windows supports long file names, maximum of 255 characters. It also allowed space to be used in between file name. This helps to make files and folders (directory/subdirectory) easier to organize and find.

Shortcuts

As the name suggests, SHORTCUTS are the shortest way to access programs, files and other resources in Windows. Instead of going through the structural process of accessing a program, one can create "shortcuts" to access them. It creates links for easy access to files, programs, folders, and more.

Multitasking

Multitasking allows the user to activate and accomplish more than one task at a time. For example, work on a document file in WORD programs, while copying file from other computer available in the network. With Windows , 32-bit computing environment, the user can do more than one task a time.

Easy Internet Access

One of the most useful and entirely new features is Window’s easy access to Internet. It provides built-in Internet functionality to setup link and access Internet with less number of Hardware and Software requirement. It also provides connectivity software from Microsoft Network (MSN), CompuServe, and America Online. It also improves the efficiency of working on Internet with applications that support the latest Internet technologies, such as ActiveX, Java, and streaming audio and video.

Software Compatibility

Windows OS provides complete backward compatibility. It is easily compatible with applications developed for MS-DOS and other version’s of Window environment. It also supports the latest 32-bit technology. Most of the latest software packages are now built on Windows operating environment.

Great Gaming Platform

Windows support rich graphics, high quality audio and video. It also requires all most no editing of *Config.sys* files so as to support these technologies. All this is possible because of Windows 95 compatibility with latest and hottest technologies like Plug and Play, AutoPlay, and built-in support for MIDI and digital and surrounds video.

Hardware Compatibility

Windows provides greater Hardware compatibility as compare to any other operating environment. It has flexibility of supporting hardware from different vendors. Its Plug and Play functionality allows to insert the hardware card into the computer and when the computer is turned on Windows automatically recognizes and sets up the hardware.

Find Utility

Find Utility of Windows allows you to do searches by partial name, last modified date, or full text. In addition, you can save, rename, or view files from within the result pane, just like you can from Windows Explorer.

Help

Windows provides online help to accomplish a task. If the user is not sure how to perform a task, Windows Help will provide structured process how to accomplish the task. Simply right-click on any object in the user interface (icons) and you'll get relevant descriptions about that object.

Lesson Number: 5 ***Writer: Vishal Verma***

Computer and Communications ***Vetter:***

5.1 OBJECTIVE

The objective of this lesson is to make the students familiar with the basics of Computer Network . After studying this lesson they will be familiar with:

1. Type of computer single user, multi user, client server etc?
2. Concept of Network and network protocols?
3. Basics of Mosaic, Gopher, facility of WWW etc?

5.2 SINGLE USER

Single user means the system on which at most only user can execute his/her processes. A system is identified as single user either by using the hardware available for use or by observing the operating system installed on it. A simple stand alone system if not connected with other computers through networking, and is used to fulfill the requirements of only user then by default it is considered as single user system. On the other hand same machine if used to fulfill the requirement of more than one user either thorough networking or by taking request of more than one user at the same time. Now- a - days almost all the system either from hardware point of view from operating system point are the multiuser systems. In the early days the operating systems like DOS make the system single user.

5.3 MULTI USER

Multi user system is the system which allows more than one users to fulfill his/her requirements from the system at the same time. Generally the operating system available in the market allows multiple users to complete their requirements from the system. Availability of algorithms to manage the multiple requests on the same machine also makes the system to behave as multi users system. Multiuser systems are supposed to be connected through each other via networking or internet to take and execute the requests of all the users. The concept of parallel computing is one of the best example of multiuser systems.

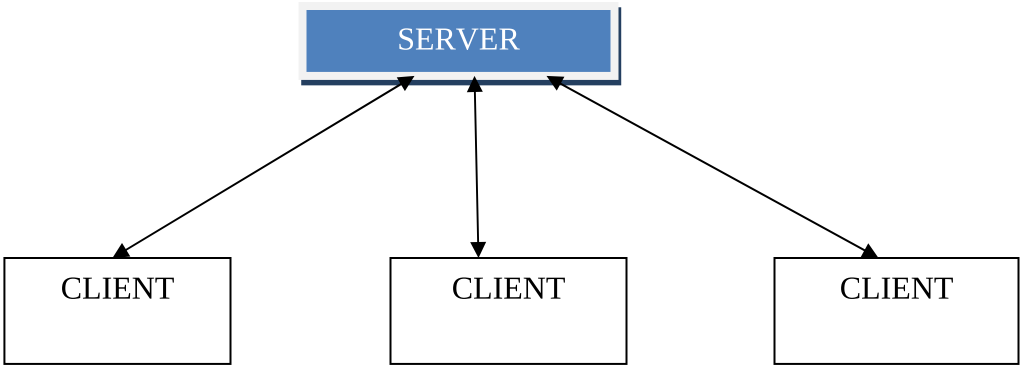
Operating system like windows, unix , linux all are the multiuser operating systems.

5.4 WORK STATION

Workstation is the term used to describe any system on which user can submit his request and expect the result of execution. In case of systems connected through each other via networking, all the connected systems are considered as workstation. If one system bears some data which can fulfill the requests of more than one users than we can call that system as server and rest all can be called as workstations for different users. Technology allow the user even to make the system behave as workstations as per the requirements of the user like in a networking lab one system may be reserved to behave as workstation to take printout and another system may be reserved to behave as workstation to use internet like facility by the user of the organization. In technical terms workstation is used to refer to a system which provide the means to fulfill the needs of the employee as per requirements.

5.5 CLIENT SERVER SYSTEMS

This system is used in an environment in which the data is supposed to be shared among a large group of users. To make the data available to all the user at the same time or as per their requirements this approach is always considered as one of the best alternate. In this approach the data to be shred is stored on the single system known as the server. We can define the server as the system which contains some data that can be demanded by any other system as per requirement, i.e the system which can fulfill the requirements of any other system/user is known as server. The client is the system which in itself is not rich in terms of available hardware and software, and is dependent on other system to complete its basic tasks time to time. The clients are the workstations where the users submit their requests. On receiving requests from the user the client system transfer the request in the formal format to the server on which the required data is stored. All the calculations are done at the server end; data is also stored on the server. Hence the client systems have fewer requirements in terms of required hardware and software. On the other hand the server systems are required to handle the requests of all the clients and hence must be rich in available hardware and software combinations. The client server system can be represented with a diagram as shown below.



CLIENT – SERVER ARCHITECTURE

5.6 COMPUTER NETWORKS

Networking is the field of computers science which deals with the principles and rules for connecting the systems together. Systems are connected to share the data stored among them and to make the communication possible for the user who are far away from each other and still want to share his ideas. Network is the need of the today's world, it's almost impossible to imagine world without network. The word "Network" tells the basics of how we can connect the systems together, it decide all the rules (protocols) for connecting the systems and to for transfer of data from one machine to other. Under these rules the network administrator decides the rule for naming and addressing of the machines which are considered for networking. Networking can be thought to be implemented at small as well as large scale, e.g it can be implemented in lab, in office, in city or among the countries as well. Depending upon the scale upto which we want to implement the network we can refer them by different names like LAN ,MAN ,WAN etc

5.6.1 NETWORK PROTOCOLS

Network protocols are the rules and regulations which are used to establish a formal method of communication among the systems; participate in the networking. A number of protocols are designed for various purposes like: http.ftp, smtp, dhcp, mtp etc. They play important role when user has are of different type in terms of operating system and available hardware. The number of bits used to represent the data may be different or same in number and protocols play very important role to manage the contradictions like this at the time of data transfer. In the network architecture like ISO/OSI, a number of protocols are used at different layers of the architecture. Each protocols plays very important role in refining the data at one layer for use in the next layer. To show the importance of tasks completed by a protocol some time additional headers are also added with the data frame to be transferred to the next layer. IEEE also specify a number of rules which must be followed to establish the networking like LAN MAN etc. These standards are also known as the protocols. Some of the well known protocols for establishing network are IEEE 802.10 IEEE 802.11 etc. In protocols like this specification like maximum length of network cable segment, maximum number of systems that can be connected, maximum achievable data transfer rate etc are described.

5.6.2 LAN and WAN

A local area network (LAN) is a computer network that connects computers and devices in a limited geographical area such as home, school, computer laboratory or office building. The defining characteristics of LAN include their usually higher data-transfer rates, smaller geographic area, and lack of a need for leased telecommunication lines. ARCNET, Token Ring and other technologies have been used in the past, but Ethernet over twisted pair cabling and Wi-Fi are the two most common technologies currently in use to establish the LAN. In its simplest form the LAN is established by connecting together the more than one systems over a small geographical area. Generally a system (powerful than others) among the group is setup to behave like the server for all other systems. This system rule all other systems in the networking, in terms of keeping their data safe, establishing a break-less communication among them and to fulfill requirements of all the users. For small area communication among the systems, LAN is one of the best alternate.

WANs are used to connect LANs and other types of networks together, so that users and computers in one location can communicate with users and computers in other locations (geographically remote). Many WANs are built for one particular organization and are private. Others, built by Internet service providers, provide connections from an organization's LAN to the Internet. WANs are often built using leased lines. At each end of the leased line, a router connects to the LAN on one side and a hub within the WAN on the other. Leased lines can be very expensive. Instead of using leased lines, WANs can also be built using less costly circuit switching or packet switching methods. Network protocols including TCP/IP deliver transport and addressing functions. Protocols including Packet over SONET/SDH, MPLS, ATM and Frame relay are often used by service providers to deliver the links that are used in WANs. X.25 was an important early WAN protocol, and is often considered to be the "grandfather" of Frame Relay as many of the underlying protocols and functions of X.25 are still in use today (with upgrades) by Frame Relay.

There are a number of alternates to establish the connections in WAN like using leased lines, circuit switching, packet switching and cell relay.

5.7 INTERNET FACILITY THROUGH WWW

The World Wide Web, abbreviated as WWW and commonly known as the Web, is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them by via hyperlinks. Using concepts from earlier hypertext systems, English engineer and computer scientist Sir Tim Berners-Lee, now the Director of the World Wide Web Consortium, wrote a proposal in March 1989 for what would eventually become the World Wide Web. At CERN in Geneva, Switzerland, Berners-Lee and Belgian computer scientist Robert Cailliau proposed in 1990 to use "HyperText ... to link and access information of various kinds as a web of nodes in which the user can browse at will", and publicly introduced the project in December.

"The World-Wide Web (W3) was developed to be a pool of human knowledge, and human culture, which would allow collaborators in remote sites to share their ideas and all aspects of a common project."

Linking

Graphic representation of a minute fraction of the WWW, demonstrating hyperlinks. Over time, many web resources pointed to by hyperlinks disappear, relocate, or are replaced with different content. This makes hyperlinks obsolete, a phenomenon referred to in some circles as link rot and the hyperlinks affected by it are often called dead links. The ephemeral nature of the Web has prompted many efforts to archive web sites. The Internet Archive, active since 1996, is one of the best-known efforts.

Dynamic updates of web pages

JavaScript is a scripting language that was initially developed in 1995 by Brendan Eich, then of Netscape, for use within web pages. The standardized version is ECMAScript. To overcome some of the limitations of the page-by-page model described above, some web applications also use Ajax (asynchronous JavaScript and XML). JavaScript is delivered with the page that can make additional HTTP requests to the server, either in response to user actions such as mouse-clicks, or based on lapsed time. The server's responses are used to modify the current page rather than creating a new page with each response. Thus the server only needs to provide limited, incremental information. Since multiple Ajax requests can be handled at the same time, users can interact with a page even while data is being retrieved. Some web applications regularly poll the server to ask if new information is available.

Privacy

Computer users, who save time and money, and who gain conveniences and entertainment, may or may not have surrendered the right to privacy in exchange for using a number of technologies including the Web. Worldwide, more than a half billion people have used a social network service, and of Americans who grew up with the Web, half created an online profile and are part of a generational shift that could be changing norms. Facebook progressed from U.S. college students to a 70% non-U.S. audience, and in 2009 estimated that only 20% of its members use privacy settings. In 2010 (six years after co-founding the company), Mark Zuckerberg wrote, "we will add privacy controls that are much simpler to use".

Security

The Web has become criminals' preferred pathway for spreading malware. Cybercrime carried out on the Web can include identity theft, fraud, espionage and intelligence gathering. Web-based vulnerabilities now outnumber traditional computer security concerns, and as measured by Google, about one in ten web pages may contain malicious code. Most Web-based attacks take place on legitimate websites, and most, as measured by Sophos, are hosted in the United States, China and Russia. The most common of all malware threats is SQL injection attacks against websites. Through HTML and URIs the Web was vulnerable to attacks like cross-site scripting (XSS) that came with the introduction of JavaScript and were exacerbated to some degree by Web 2.0 and Ajax web design that favors the use of scripts. Today by one estimate, 70% of all websites are open to XSS attacks on their users.

Standards

Many formal standards and other technical specifications and software define the operation of different aspects of the World Wide Web, the Internet, and computer information exchange. Many of the documents are the work of the World Wide Web Consortium (W3C), headed by Berners-Lee, but some are produced by the Internet Engineering Task Force (IETF) and other organizations.

Usually, when web standards are discussed, the following publications are seen as foundational:

- Recommendations for markup languages, especially HTML and XHTML, from the W3C. These define the structure and interpretation of hypertext documents.
- Recommendations for stylesheets, especially CSS, from the W3C.
- Standards for ECMAScript (usually in the form of JavaScript), from Ecma International.
- Recommendations for the Document Object Model, from W3C.

Internationalization

The W3C Internationalization Activity assures that web technology will work in all languages, scripts, and cultures. Beginning in 2004 or 2005, Unicode gained ground and eventually in December 2007 surpassed both ASCII and Western European as the Web's most frequently used character encoding. Originally RFC 3986 allowed resources to be identified by URI in a subset of US-ASCII. RFC 3987 allows more characters—any character in the Universal Character Set—and now a resource can be identified by IRI in any language

5.8 MOSAIC

Mosaic is the web browser credited with popularizing the World Wide Web. It was also a client for earlier protocols such as FTP, NNTP, and gopher. Its clean, easily understood user interface, reliability, Windows port and simple installation all contributed to making it the application that opened up the Web to the general public. Mosaic was also the first browser to display images in line with text instead of displaying images in a separate window. While often described as the first graphical web browser, Mosaic was preceded by the lesser-known Erwise and ViolaWWW. Mosaic was developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign beginning in late 1992. NCSA released the browser in 1993, and officially discontinued development and support on January 7, 1997. However, it can still be downloaded from NCSA. Fifteen years after Mosaic's introduction, the most popular contemporary browsers, Internet Explorer , Mozilla

Firefox and Google Chrome retain many of the characteristics of the original Mosaic graphical user interface (GUI) and interactive experience. Netscape Navigator was later developed by James H. Clark and many of the original Mosaic authors; however, it intentionally shared no code with Mosaic. Netscape Navigator's code descendant is Mozilla.

Mosaic was not the first web browser for Microsoft Windows; this was Tom Bruce's little-known Cello. The UNIX version of Mosaic was already making it famous before the Windows and Mac versions came out. Other than displaying images embedded in the text rather than in a separate window, Mosaic did not in fact add many features to the browsers it was modeled on, like ViolaWWW. But Mosaic was the first browser written and supported by a team of full-time programmers, which was reliable and easy enough for novices to install, and the inline graphics proved immensely appealing. Mosaic made the Web accessible to the ordinary person for the first time.

Reid also refers to Matthew Gray's well-respected website, Internet Statistics: Growth and Usage of the Web and the Internet, which indicates a dramatic leap in web use around the time of Mosaic's introduction.

By 1998 its user base had almost completely evaporated, being replaced by other web browsers. After NCSA stopped work on Mosaic, development of the NCSA Mosaic for the X Window System source code was continued by several independent groups. These independent development efforts include mMosaic (multicast Mosaic) which ceased development in early 2004, and Mosaic-CK and VMS Mosaic which are both under active development as of July 2010

5.9 GOPHER

The original Gopher system was released in late spring of 1991 by Mark McCahill, Farhad Anklesaria, Paul Lindner, Daniel Torrey, Adam Huminsky, and Bob Alberti of the University of Minnesota. Its central goals were, A file-like hierarchical arrangement that would be familiar to users, simple syntax, system that can be created quickly and inexpensively, Extending the file system metaphor, such as searches. Gopher combines document hierarchies with collections of services, including WAIS, the Archie and Veronica search engines, and gateways to other information systems such as FrTP and Usenet.

The general interest in Campus-Wide Information Systems (CWISs) in higher education at the time, and the ease with which a Gopher server could be set up to create an instant CWIS with links to other sites' online directories and resources were the factors contributing to Gopher's rapid adoption. By 1992, the standard method of locating someone's e-mail address was to find their organization's CCSO name-server entry in Gopher, and query the name-server.

Various origins are possible for the name. The University of Minnesota mascot is the gopher, a gofer (same sound) is an assistant who "goes for" things, and a gopher browser through the ground to reach a desired location.

The Gopher protocol (/□go□fər/) is a TCP/IP Application layer protocol designed for distributing, searching, and retrieving documents over the Internet. Strongly oriented towards a menu-document design, the Gopher protocol was a predecessor of (and later, an alternative to) the World Wide Web.

The protocol offers some features not natively supported by the Web and imposes a much stronger hierarchy on information stored on it. Its text menu interface is well-suited to computing environments that rely heavily on remote text-oriented computer terminals, which were still common at the time of its creation in 1991, and the simplicity of its protocol facilitated a wide variety of client implementations.

Gopher browser plugins

For Mozilla Firefox and SeaMonkey, OverbiteFF extends Gopher browsing and supports Firefox 4. It includes support for accessing Gopher servers not on port 70 using a whitelist and for CSO/ph queries, and allows versions of Firefox and SeaMonkey that do not support Gopher natively to access Gopher servers. Plugins are also available for Konqueror and a proxy-based extension for Google Chrome.

Gopher clients for mobile devices

Some have suggested that the bandwidth-sparing simple interface of Gopher would be a good match for mobile phones and Personal digital assistants (PDAs), but so far, mobile adaptations of HTML and XML and other simplified content have proved more popular. The PyGopherd server provides a built-in WML front-end to Gopher sites served with it. An application for Android 1.5+ is in development and was released in alpha stage. A Java ME client is also available for compatible devices.

Other Gopher clients

Gopher was at its height of popularity during a time when there were still many equally competing computer architectures and operating systems. As such, there are several Gopher clients available for Acorn RISC OS, AmigaOS, Atari MiNT, CMS, DOS, classic Mac OS, MVS, NeXT, OS/2 Warp, most UNIX-like operating systems, VMS, Windows 3.x, and Wirndows 9x. GopherVR was a client designed for 3D visualization, and there is even a Gopher client MOO object. The majority of these clients are hard coded to work on TCP port 70.

Gopher to HTTP gateways

Users of Web browsers that have incomplete or no support for Gopher can access content on Gopher servers via a server gateway or proxy server that converts Gopher menus into HTML; one such proxy is the Floodgap Public Gopher Proxy. Similarly, certain server packages such as GN and PyGopherd have built-in Gopher to HTTP interfaces.

5.10 HTML

HTML, which stands for HyperText Markup Language, is the predominant markup language for web pages. A markup language is a set of markup tags, and HTML uses markup tags to describe web pages. HTML is written in the form of HTML elements consisting of "tags" surrounded by angle brackets (like <html>) within the web page content. HTML tags normally come in pairs like <b> and </b>. The first tag in a pair is the *start tag*, the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). The purpose of a web browser is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML web pages. HTML can also be used to include Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both HTML and CSS standards, encourages the use of CSS over explicit presentational markup.

HTML markup consists of several key components, including *elements* (and their *attributes*), character-based *data types*, *character references* and *entity references*. Another important component is the *document type declaration*, which specifies the Document Type Definition. As of HTML 5, no Document Type Definition will need to be specified and will only determine the layout mode.

HTML documents are composed entirely of HTML elements that, in their most general form have three components: a pair of element *tags*, a "start tag" and "end tag"; some element *attributes* within the start tag; and finally, any textual and graphical *content* between the start and end tags. The HTML element is everything between and including the tags. Each tag is enclosed in angle brackets.

The general form of an HTML element is therefore: <tag attribute1="value1" attribute2="value2">*content to be rendered*</tag> The name of the HTML element is also the name of the tag. Note that the end tag's name is preceded by a slash character, "/". If attributes are not assigned, default values are used.

The Hello world program, a common computer program employed for comparing programming languages, scripting languages and markup languages is made of 9 lines of code although in HTML newlines are optional:

```
<!doctype html>
```

```
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

(The text between <html> and </html> describes the web page, and The text between <body> and </body> is the visible page content.)

There are several types of markup elements used in HTML.

Structural markup describes the purpose of text. For example, <h2>Golf</h2> establishes "Golf" as a second-level heading, which would be rendered in a browser in a manner similar to the "HTML markup" title at the start of this section. Structural markup does not denote any specific rendering, but most web browsers have default styles for element formatting. Text may be further styled with Cascading Style Sheets (CSS).

Presentational markup describes the appearance of the text, regardless of its purpose. For example <b>boldface</b> indicates that visual output devices should render "boldface" in bold text, but gives little indication what devices which are unable to do this (such as aural devices that read the text aloud) should do. In the case of both <b>bold</b> and <i>italic</i>, there are other elements that may have equivalent visual renderings but which are more semantic in nature, such as <strong>strong emphasis</strong> and <em>emphasis</em> respectively.

Hypertext markup makes parts of a document into links to other documents. An anchor element creates a hyperlink in the document with the href attribute set to the link URL. To render an image as a hyperlink, an 'img' element is inserted as content into the 'a' element. Like 'br', 'img' is an empty element with attributes but no content or closing tag. <a href="http://example.org"><img src="image.gif" alt="descriptive text" width="50" height="50"></a>.

Attributes

Most of the attributes of an element are name-value pairs, separated by "=" and written within the start tag of an element after the element's name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML). Leaving attribute values unquoted is considered unsafe. In contrast with name-value pair attributes, there are some attributes that affect the element simply by their presence in the start tag of the element, like the ismap attribute for the img element

Data types

HTML defines several data types for element content, such as script data and stylesheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

5.11 ELEMENTS OF JAVA

The syntax of a programming language tells you what code it is possible to write—what the machine will understand. Style tells you what you ought to write—what the humans reading the code will understand. Code written with a consistent, simple style will be maintainable, robust, and contain fewer bugs. Code written with no regard to style will contain more bugs. It may simply be thrown away and rewritten rather than maintained.

Keywords

There are certain words with a specific meaning in java which tell (help) the compiler what the program is supposed to do. These Keywords cannot be used as variable names, class names, or method names. Keywords in java are case sensitive, all characters being lower case.

Keywords are reserved words that are predefined in the language All the keywords are in lowercase.

abstract
boolean
break
byte
case
catch
char
class
const
continue
default
do
double
else
extends
final
finally
float
for
goto
if
implements
import
instanceof
int
interface
long
native
new
package
private
protected
public
return
short
static
strictfp
super
switch
synchronized
this
throw
throws
transient
try
void
volatile
while

Comments

Comments are descriptions that are added to a program to make code easier to understand. The compiler ignores comments and hence its only for documentation of the program. Java supports three comment styles.

Block style comments begin with /* and terminate with */ that spans multiple lines.

Line style comments begin with // and terminate at the end of the line. (Shown in the above program)

Documentation style comments begin with /** and terminate with */ that spans multiple lines. They are generally created using the automatic documentation generation tool, such as javadoc.

Variable, Identifiers and Data Types

Variables are used for data that change during program execution. All variables have a name, a type, and a scope. The programmer assigns the names to variables, known as identifiers. An Identifier must be unique within a scope of the Java program. Variables have a data type, that indicates the kind of value they can store. Variables declared inside of a block or method are called local variables; They are not automatically initialized. The compiler will generate an error as a result of the attempt to access the local variables before a value has been assigned.

The **data type** indicates the attributes of the variable, such as the range of values that can be stored and the operators that can be used to manipulate the variable. Java has four main primitive data types built into the language. You can also create your own composite data types.

Java has four main primitive data types built into the language. We can also create our own data types.

- ***Integer:*** byte, short, int, and long.
- ***Floating Point:*** float and double
- ***Character:*** char
- ***Boolean:*** variable with a value of true or false.

Classes

A class is nothing but a blueprint for creating different objects which defines its properties and behaviors. An object exhibits the properties and behaviors defined by its class. A class can contain fields and methods to describe the behavior of an object. Methods are nothing but members of a class that provide a service for an object or perform some business logic.

Objects

An object is an instance of a class created using a new operator. The new operator returns a reference to a new instance of a class. This reference can be assigned to a reference variable of the class. The process of creating objects from a class is called instantiation. An object reference provides a handle to an object that is created and stored in memory. In Java, objects can only be manipulated via references, which can be stored in variables.

Interface

An Interface is a contract in the form of collection of method and constant declarations. When a class implements an interface, it promises to implement all of the methods declared in that interface.

Instance Members

Each object created will have its own copies of the fields defined in its class called instance variables which represent an object’s state. The methods of an object define its behavior called instance methods. Instance variables and instance methods, which belong to objects, are collectively called instance members. The dot ‘.’ notation with a object reference is used to access Instance Members.

Static Members

Static members are those that belong to a class as a whole and not to a particular instance (object). A static variable is initialized when the class is loaded. Similarly, a class can have static methods. Static variables and static methods are collectively known as static members, and are declared with a keyword static. Static members in the class can be accessed either by using the class name or by using the object reference, but instance members can only be accessed via object references.

Lesson Number: 6 ***Writer: Vishal Verma***

INFORAMATION INTEGRITY DEFINITE ***Vetter:***

6.1 OBJECTIVE

The objective of this lesson is to make the students familiar with the basics of integrity and security aspects related with computers. After studying this lesson they will be familiar with:

1. Integrity and its related concepts?
2. How to maintain integrity among data in computers?
3. Issues related with computer security, its components and preventive measures?

6.2 INTRODUCTION

Information is very important term for all of the computer as well as non computer user. As per its basic definition it can be defined as a processed data, i.e. the data on which all the desired operation has been performed to get the results or reports as per users requirements. Though the information is important for us yet the data from which it is abstracted has its own important and collection and storage in best way should be maintained. The integrity reflects the unit or collection of data in a single place so that all other aspects like security, updation and maintenance can be done in better way. In the text discussed below the main concentration will remain on the definition of integrity and the steps to be taken to main integrity and security in the systems.

6.3 ENSURING INTEGRITY

Integrity among the information is so important that we cannot imagine any business of system to sustain well without taking proper steps to maintain integrity among the data or information used in that system. Only on keeping the Information Integrity you can expect a solution that scales to meet the needs of an information-intensive Fortune 500 business and standardize controls that eliminate errors throughout the whole enterprise – from customer quotation to SEC filings. The technology landscape is awash with buzzwords and terminology used without common understanding and proper definitions. The exact meaning of Data Quality can vary widely from person to person. The term has been used by a number of different companies to mean things like address standardization, data profiling, and even product catalog synchronization, but one common findings among all of these definitions is the concept of data integrity since to achieve any of the definition it is necessary to keep the whole data as integrated in the storage. The phrase “Information Integrity” to used to describe the solution for many problems because it carries a more precise definition of a concept that is an essential component to enhancing the value of information in the organization. In an information-intensive business, this means enhancing the value of enterprise. Information Integrity is a prerequisite for many other information management initiatives. If the underlying information isn’t of a sufficient level of integrity, the success of business activities relying on the information will be limited. Before starting on any Data Warehouse, Business Intelligence (BI) project, or Business Activity Monitoring (BAM) project, it is must to take necessary steps for ensuring integrity.

Information Integrity is the trustworthiness and dependability of information. More specifically, it is the accuracy, consistency and reliability of the information content, processes and systems.
A distinguishing feature of Information Integrity is its focus on all of the environments or domains that ultimately govern the integrity of information. The issue is not simply "bad data," but the underlying systems and processes that produce unreliable or inaccurate content. The three domains of Information Integrity are: content, process, and system. The integrated relationship among these three domains is the overall context of Information Integrity.

- **Information Content** is the set of data elements (or groups) provided to the user(s) to enable the attainment of a task's objectives. The content includes many forms, e.g., numeric, text, spreadsheets, business reports, databases, and more.

- A **Process** is the organized set of logical functions designed to transform an input into a specified output. Examples of processes include claims processing and financial reporting.

- A **System** is the organized set of physical and logical components (human, electronic, mechanical, or other) configured to achieve a specific purpose. Examples of systems include computer applications, organizational units, and even governments.

6.4 COMPUTER SECURITY

Computer security is a branch of computer technology known as information security as applied to computers and networks. The objective of computer security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The term computer system security means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The strategies and methodologies of computer security often differ from most other computer technologies because of its somewhat elusive objective of preventing unwanted computer behavior instead of enabling wanted computer behavior. The technologies of computer security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behavior.

There are 4 approaches to security in computing; sometimes a combination of approaches is valid:

1. Trust all the software to abide by a security policy but the software is not trustworthy (this is computer insecurity).

2. Trust all the software to abide by a security policy and the software is validated as trustworthy (by tedious branch and path analysis for example).

3. Trust no software but enforce a security policy with mechanisms that are not trustworthy (again this is computer insecurity).

4. Trust no software but enforce a security policy with trustworthy hardware mechanisms.

Computers consist of software executing atop hardware, and a "computer system" is, by frank definition, a combination of hardware, software (and, arguably, firmware, should one choose so separately to categorize it) that provides specific functionality, to include either an explicitly expressed or (as is more often the case) implicitly carried along security policy. Indeed, citing the Department of Defense Trusted Computer System Evaluation Criteria (the TCSEC, or Orange Book)—archaic though that may be —the inclusion of specially designed hardware features, to include such approaches as tagged architectures and (to particularly address "stack smashing" attacks of recent notoriety) restriction of executable text to specific memory regions and/or register groups, was a *sine qua non* of the higher evaluation classes, to wit, B2 and above.) There are various strategies and techniques used to design security systems. However there are few, if any, effective strategies to enhance security after design. One technique enforces the principle of least privilege to great extent, where an entity has only the privileges that are needed for its function. That way even if an attacker gains access to one part of the system, fine-grained security ensures that it is just as difficult for them to access the rest.

Security classification for information

An important aspect of information security and risk management is recognizing the value of information and defining appropriate procedures and protection requirements for the information. Not all information is equal and so not all information requires the same degree of protection. This requires information to be assigned a security classification.

The first step in information classification is to identify a member of senior management as the owner of the particular information to be classified. Next, develop a classification policy. The policy should describe the different classification labels, define the criteria for information to be assigned a particular label, and list the required security controls for each classification.

Some factors that influence which classification information should be assigned include how much value that information has to the organization, how old the information is and whether or not the information has become obsolete. Laws and other regulatory requirements are also important considerations when classifying information.

The type of information security classification labels selected and used will depend on the nature of the organization, with examples being:

- In the business sector, labels such as: Public, Sensitive, Private, Confidential.
- In the government sector, labels such as: Unclassified, Sensitive But Unclassified, Restricted, Confidential, Secret, Top Secret and their non-English equivalents.
- In cross-sectoral formations, the Traffic Light Protocol, which consists of: White, Green, Amber and Red.

All employees in the organization, as well as business partners, must be trained on the classification schema and understand the required security controls and handling procedures for each classification. The classification a particular information asset has been assigned should be reviewed periodically to ensure the classification is still appropriate for the information and to ensure the security controls required by the classification are in place.

6.5 PERVERSE SOFTWARE

It is a program which is causing hindrances of other program execution such a way resulting in modification or complete destruction of data without the user's intention or even sabotaging the operational system.

6.6 CONCEPT AND COMPONENT OF SECURITY

The word security refer to the implementation of some methods by using which we can make the system or data to be used by only authorized persons only. If these measures are not implemented in well manner than the intruders may cause use our data for their benefits and cause us a lot of loss in terms of money and wealth. The basic component on which security issues must be applied are : System and Data or Information. Here we are discussing both aspects of security and their components.

System security : Computer security can focus on ensuring the availability and correct operation of a computer system without concern for the information stored or processed by the computer.

Governments, military, corporations, financial institutions, hospitals, and private businesses amass a great deal of confidential information about their employees, customers, products, research, and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers. Should confidential information about a business' customers or finances or new product line fall into the hands of a competitor, such a breach of security could lead to lost business, law suits or even bankruptcy of the business. Protecting confidential information is a business requirement, and in many cases also an ethical and legal requirement.

Information Security : **Information security** means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. The field of information security has grown and evolved significantly in recent years. There are many ways of gaining entry into the field as a career. It offers many areas for specialization including: securing network(s) and allied infrastructure, securing applications and databases, security testing, information systems auditing, business continuity planning and digital forensics science, etc.

The terms information security, computer security and information assurance are frequently incorrectly used interchangeably. These fields are interrelated often and share the common goals of protecting the confidentiality, integrity and availability of information; however, there are some subtle differences between them. These differences lie primarily in the approach to the subject, the methodologies used, and the areas of concentration. Information security is concerned with the confidentiality, integrity and availability of data regardless of the form the data may take: electronic, print, or other forms.

Basic Principles

Key concepts

For over twenty years, information security has held confidentiality, integrity and availability (known as the CIA triad) to be the core principles of information security.

There is continuous debate about extending this classic concept. Other principles such as Accountability have sometimes been proposed for addition - it has been pointed out that issues such as Non-Repudiation do not fit well within the three core concepts, and as regulation of computer systems has increased (particularly amongst the Western nations) Legality is becoming a key consideration for practical security installations.

In 2002, Donn Parker proposed an alternative model for the classic CIA triad that he called the six atomic elements of information. The elements are confidentiality, possession, integrity, authenticity, availability, and utility. The merits of the Parkerian hexad are a subject of debate amongst security professionals.

Confidentiality

Confidentiality is the term used to prevent the disclosure of information to unauthorized individuals or systems. For example, a credit card transaction on the Internet requires the credit card number to be transmitted from the buyer to the merchant and from the merchant to a transaction processing network. The system attempts to enforce confidentiality by encrypting the card number during transmission, by limiting the places where it might appear (in databases, log files, backups, printed receipts, and so on), and by restricting access to the places where it is stored. If an unauthorized party obtains the card number in any way, a breach of confidentiality has occurred.

Breaches of confidentiality take many forms. Permitting someone to look over your shoulder at your computer screen while you have confidential data displayed on it could be a breach of confidentiality. If a laptop computer containing sensitive information about a company's employees is stolen or sold, it could result in a breach of confidentiality. Giving out confidential information over the telephone is a breach of confidentiality if the caller is not authorized to have the information.

Confidentiality is necessary (but not sufficient) for maintaining the privacy of the people whose personal information a system holds.

Integrity

In information security, integrity means that data cannot be modified undetectably. This is not the same thing as referential integrity in databases, although it can be viewed as a special case of Consistency as understood in the classic ACID model of transaction processing. Integrity is violated when a message is actively modified in transit. Most cipher systems provide message integrity along with privacy as part of the encryption process. Messages that have been tampered with in flight will not decrypt successfully.

Availability

For any information system to serve its purpose, the information must be available when it is needed. This means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly. High availability systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks.

Authenticity

In computing, e-Business and information security it is necessary to ensure that the data, transactions, communications or documents (electronic or physical) are genuine. It is also important for authenticity to validate that both parties involved are who they claim they are.

Non-repudiation

In law, non-repudiation implies one's intention to fulfill their obligations to a contract. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction.

Electronic commerce uses technology such as digital signatures and encryption to establish authenticity and non-repudiation.

6.7 PREVENTIVE MEASURES AND TREATMENT

Measures to enhance computer security

Security and systems design

Most current real-world computer security efforts focus on external threats, and generally assume that the computer system itself is not vulnerable. This, according to experts is considered to be a disastrous mistake, and point out this to be the cause of much of the insecurity of current computer systems. Once an attacker has accessed one part of a system without fine-grained security, he ultimately has access to other parts of the system.

Serious financial damage has been caused by computer security breaches on the computer system. Individuals who have had computer systems infected with spy ware or malware bear costly and time-consuming processes in cleaning. Spy ware is considered to be a problem specific to the various Microsoft Windows operating systems.

Security measures

There are various measures employed to enhance system security:

- User account access controls and cryptography: These can protect systems files and data, respectively. This is by use of log-ins and passwords.

• Firewalls: These are the most common prevention systems from a network security perspective as they can (if properly configured) shield access to internal network services, and block certain kinds of attacks through packet filtering

• Intrusion Detection Systems (IDS's): These are systems that are designed to detect network attacks in progress and assist in post-attack forensics, while audit trails and logs serve a similar function for individual systems such as keeping track of all those who have been in the system.

Today, computer security comprises mainly "preventive" measures, like firewalls or an Exit Procedure. A firewall can be defined as a way of filtering network data between a host or a network and another network, such as the Internet and is normally implemented as software running on the machine, hooking into the network stack.

Use a good antivirus program. This is the most important piece of work in preventive maintenance. Installing the antivirus program is not good enough. You should do following as well.

Set-up the program to download and install updates automatically.

Schedule periodic full-system scans.

Check the virus definitions date regularly and see whether it is up to date.

Set-up your PC to Download and install “Windows Updates” automatically. Windows updates include Operating System patches for bugs and PC security related issues. These patches can reduce many unknown computer problems.

Install anti-Spyware program to detect Spyware tools.

Install a Personal Firewall. Most of the antivirus programs are bundled with a Personal Firewall these days.

Do not download and install unknown software from Internet. This is the biggest mistake most of the PC users are doing. This software can damage the Windows registry, which causes a lot of errors.

Be very careful when you download music from the Internet. Always stick to one trustworthy website.

Perform Scandisk periodically to check the Hard Drive.

Delete temporary Internet files.

Lesson Number: 7 ***Writer: Vishal Verma***

RANGE OF APPLICATIONS ***Vetter:***

7.1 OBJECTIVE

The objective of this lesson is to make the students familiar with the basics of Scientific application and role of computer in them. After studying this lesson they will be familiar with:

1. Role of computer in Scientific applications?
2. Importance of scientific applications and their basic working principal?
3. Application of scientific applications?

7.2 INTRODUCTION

Information technology is growing in all fields of today's world. By using the basics of information technology it becomes very easy for us to develop the user friendly and more efficient applications. The uses of computer to achieve the goal in our day to day life make the things very easy to formulate and use. Now we can do the implementation of all thinkable activities very easily. Here we are discussing some of the important fields which are using the information technology for implementing the applications.

7.3 SCIENTIFIC APPLICATIONS

An application that simulates real-world activities using mathematics. Real-world objects are turned into mathematical models and their actions are simulated by executing the formulas.
For example, some of an airplane's flight characteristics can be simulated in the computer. Rivers, lakes and mountains can be simulated. Virtually any objects with known characteristics can be modeled and simulated.
Simulations use enormous calculations and often require supercomputer speed. As personal computers became more powerful, more laboratory experiments have been converted into computer models that can be interactively examined by students without the risk and cost of the actual experiments. All of the above discussed examples are no doubt very important for all of us, since they affect our day to day life directly or indirectly. The implementation of scientific

applications can’t be imagined without the use of computers of information technology. The sharing of data and thoughts among the group of people is become possible just because of information technology. For implementation of any of the problems solutions the simulated model helps a lot in observing the after effects of the solution. We can see the possible outcomes and change in the outcomes corresponding to any change in the input parameters very easily by creating the simulated program for any scientific problem. Hence the information technology provides a very good alternate to the scientists to do the experiments with low investments and good way of observing the outputs. Nuclear reactors and their working can be implemented with some specific software which otherwise are very costly and difficult to achieve. By using the computers to solve the scientific problems make the project very effective and less expensive as compare to real implementation of the same problem.

Some of the important identified scientific applications are:

Computer animation

Computer animation is the art, technique and science of creating moving images via the use of computers. Increasingly it is created by means of 3D computer graphics, though 2D computer graphics are still widely used for stylistic, low bandwidth, and faster real-time rendering needs. Sometimes the target of the animation is the computer itself, but sometimes the target is another medium, such as film. It is also referred to as CGI (Computer-generated imagery or computer-generated imaging), especially when used in films.

In the natural sciences

Star formation: The featured plot is a Volume plot of the logarithm of gas/dust density in an Enzo star and galaxy simulation. Regions of high density are white while less dense regions are more blue and also more transparent.

Gravity waves: Researchers used the Globus Toolkit to harness the power of multiple supercomputers to simulate the gravitational effects of black-hole collisions.

Massive Star Supernovae Explosions: In the image three Dimensional Radiation Hydrodynamics Calculations of Massive Star Supernovae Explosions The DJEHUTY stellar evolution code was used to calculate the explosion of SN 1987A model in three dimensions.

Molecular rendering: VisIt's general plotting capabilities were used to create the molecular rendering shown in the featured visualization. The original data was taken from the Protein Data Bank and turned into a VTK file before rendering.

In geography and ecology

Terrain rendering: VisIt can read several file formats common in the field of Geographic Information Systems (GIS), allowing one to plot raster data such as terrain data in visualizations. The featured image shows a plot of a DEM dataset containing mountainous areas near Dunsmuir, CA. Elevation lines are added to the plot to help delineate changes in elevation.

Tornado Simulation: This image was created from data generated by a tornado simulation calculated on NCSA's IBM p690 computing cluster. High-definition television animations of the storm produced at NCSA were included in an episode of the PBS television series NOVA called "Hunt for the Supertwister." The tornado is shown by spheres that are colored according to pressure; orange and blue tubes represent the rising and falling airflow around the tornado.

Climate visualization: This visualization depicts the carbon dioxide from various sources that are advected individually as tracers in the atmosphere model. Carbon dioxide from the ocean is shown as plumes during February 1900.

Atmospheric Anomaly in Times Square In the image the results from the SAMRAI simulation framework of an atmospheric anomaly in and around Times Square are visualized.

In the applied sciences

Porsche 911 model (NASTRAN model): The featured plot contains a Mesh plot of a Porsche 911 model imported from a NASTRAN bulk data file. VisIt can read a limited subset of NASTRAN bulk data files, generally enough to import model geometry for visualization.

YF-17 aircraft Plot: The featured image displays plots of a CGNS dataset representing a YF-17 jet aircraft. The dataset consists of an unstructured grid with solution. The image was created by using a pseudocolor plot of the dataset's Mach variable, a Mesh plot of the grid, and Vector plot of a slice through the Velocity field.

City rendering: An ESRI shapefile containing a polygonal description of the building footprints was read in and then the polygons were resampled onto a rectilinear grid, which was extruded into the featured cityscape.

Inbound traffic measured: This image is a visualization study of inbound traffic measured in billions of bytes on the NSFNET T1 backbone for the month of September 1991. The traffic volume range is depicted from purple (zero bytes) to white (100 billion bytes). It represents data collected by Merit Network, Inc

7.4 BUSINESS APPLICATIONS

Business software is generally any software program that helps businesses to increase productivity or measure their productivity. The term covers a large variation of uses within the business environment, and can be categorized by using a small, medium and large matrix:

- The small business market generally consists of home accounting software, and office suites such as Microsoft Office and OpenOffice.org.
- The medium size, or SME, has a broader range of software applications, ranging from accounting, groupware, customer relationship management, human resources software, outsourcing relationship management, loan origination software, shopping cart software, field service software, and other productivity enhancing applications.

- The last segment covers enterprise level software applications, such as those in the fields of enterprise resource planning, enterprise content management (ECM), business process management and product lifecycle management. These applications are extensive in scope, and often come with modules that either add native functions, or incorporate the functionality of third-party software programs.

Types of business software tools

- Digital Dashboards - Also known as Business Intelligence Dashboards, Enterprise Dashboards, or Executive Dashboards, these are visually-based summaries of business data that show at-a-glance understanding of business conditions through metrics and Key Performance Indicators (KPIs). A very popular BI tool that has arisen in the last few years.
- Online Analytical Processing, commonly known as OLAP (including HOLAP, ROLAP and MOLAP) - a capability of some management, decision support, and executive information systems that supports interactive examination of large amounts of data from many perspectives.
- Reporting software generates aggregated views of data to keep the management informed about the state of their business.
- Data mining - extraction of consumer information from a database by utilizing software that can isolate and identify previously unknown patterns or trends in large amounts of data. There are a variety of data mining techniques that reveal different types of patterns. Some of the techniques that belong here are Statistical methods (particularly Business statistics) and Neural networks as very advanced means of analyzing data.
- Business performance management (BPM).

7.5 EDUCATIONAL APPLICATIONS

Many sophisticated educational models and tools can be imagined that could significantly advance education; there are too few applications that take full advantage of the supercomputers now on students' desks and in their backpacks. There are many ways of using computers in education. One of the most promising directions is based on powerful tools and models that could be used in many different learning contexts (Feurzeig & Roberts, 1999; Gobert & Buckley, 2000; Tinker, 1990a, 1990b). Students learn from these tools and models through guided explorations of appropriate problems and challenges. To be practical, these applications need to be embedded in an educational platform that can deliver complete learning activities online and then assess student progress as they work through these activities. Teachers and educators should be able to customize the activities and assessments, so that they can tailor the learning strategies to the needs and interests of their students. The tools, models, and platforms needed to exploit this approach are sophisticated and probably too complex to be supported commercially. A better option is to make them open source and trust that their utility will support a broad community of users that will provide ongoing support and development. Successful open source software for education will need to share these characteristics; it should consist of well-designed, highly functional packages that can serve diverse needs. Small applications that are easily replicated such as a simple function graphed will not take off. Highly specialized software, even if well designed and highly functional, will fail as open source because of the limited audience willing to provide support. There is already one example of an open source educational application that fits this profile. It fills a huge need, provides substantial functionality, and is well designed. Open ACS—an online course platform—started at MIT, morphed into a product called Ars Digita that attracted considerable funding in the late 1990's, went bust, and reemerged as open source called dotLRN. Open ACS is a sub-set of dotLRN that is used by growing group of universities worldwide and supported by a community of companies.

Open ACS solves a problem faced by almost every university worldwide and many other institutions, namely what technology to use for online courses. We started offering online courses a decade ago, first using Web pages, then waiting for a university-based package that never was completed, then using Learning Space, which was later discontinued by IBM, and finally shifting to Blackboard.

Every one of these platforms, and many others that we have investigated, has limitations that restrict the educational value of the materials delivered to students. Only when we shifted to Open ACS were we able to access the source code and make the upgrades that we needed. In so doing, we were doing just what is needed to help create a viable open source community: contributing new functionality that anyone can use, not because we felt generous, but because making these improvements were in our own self-interest. Open source educational applications can thrive if the software solves an identifiable need faced by many users. It also illustrates the difficulty commercial suppliers have in providing long-term support of large educational packages.

7.6 INDUSTIRAL APPLICATIONS

Information technology plays a primary role in industries. You'd be hard pressed to find an industry that does not heavily rely on technology initiatives to help run the operational and/or strategic aspects of the business. Technology is integrated in most, if not all, areas of business and organizations have come to depend upon technology being available. Imagine a business environment without a website, electronic cash registers, customer databases, human resource systems, electronic payroll systems (including direct deposit) and other important areas an organization needs to run. When thinking about it from this respect it is not hard to realize just how much information technology is relied upon. In addition to the day to day business needs, here are a few other ways for achieving them its better to use information technology for developing industries applications

Competitive Advantage

Integrating the latest and greatest technology is useless unless it can be used to improve the business' processes and transactions and offer a viable advantage. Technology has become increasingly significant and plays a vital role for organizations to use in acquiring a competitive advantage.

*Increase Efficiency and Speed

Information systems are fast, powerful and can sort through massive amounts of data. Computer information systems offer a high level of efficiency and organization. This makes information easier to find and store. The speed at which transactions can be completed are highly desired because machines can essentially take over the mundane aspects of running a business. All that is needed is someone to operation and/or give the commands. This frees employees up to focus on other important tasks that must be handled by the human brain.

*Strategic Planning/Detect Trends

Collecting data is an advantage because any data collected can be turned in to useful information to help in strategic planning and to predict trends. All of this information gained can be attributed to the role of information technology. Without it, this information would be pretty difficult, if not impossible to attain.

The human eye couldn't possibly sort through and determine unseen patterns, but through data mining and other ways to sort through data, various industries can discover patterns unseen previously and see trends that have emerged or use past trends to try and predict future events.

*Better Deliverables for Customers

Technology is beneficial because it helps significantly increase the level of customer service when used correctly. Again, this is an area where using technology for the sake of technology is useless. Smart managers understand the value of information technology and how it can be used to increase levels of customer service, offer faster transactional processing, increase convenience and choices, and to also provide much higher levels of personalization.

No one likes to feel like a number, but technology initiatives make it possible for even the largest of businesses to offer thousands of customers individualized attention and personalization. In the event technology stops working or a large interruption is experience, this disrupts business. It has gotten to the point that the role of information technology is so vital that other areas of operations have emerged in order to protect these valuable electronic assets. Information security is one area that is making significant growth due to the importance of securing valuable data and ensuring its availability. Many career opportunities are evolving for this field. Business continuity has become an essential component of any business plan and strategy because without running technology, a business can suffer tremendously in the event of a disruption. The role of information technology in industry has become a centralized role because there are little, if any, areas of a company that does not use some sort of information system. Many of the major segments of any given business use computers or other kind of automation. Businesses today can't afford to do without it else they'll likely lose a strong position in their respective markets.

7.7 NATIONAL LEVEL WEATHER FORCASTING

The **National Weather Service (NWS)**, once known as the **Weather Bureau**, one of the six scientific agencies that make up the National Oceanic and Atmospheric Administration (NOAA) of the United States government. It is headquartered in Silver Spring, Maryland.

The NWS is tasked with providing "weather, hydrologic, and climate forecasts and warnings, adjacent waters and ocean areas, for the protection of life and property and the enhancement of the national economy." This is done through a collection of national and regional centers, and 122 local weather forecast offices (WFOs). Since the NWS is a government agency, most of its products are in the public domain and available free of charge.

Forecasting

The NWS issues a comprehensive package of forecast products to support a variety of users, including the general public. Although text forecasts have been the primary means of product dissemination, the NWS has been converting its forecast products to a digital, gridded format. Each of the 122 Weather Forecast Offices (WFOs) send their graphical forecasts to a national server to be compiled in the National Digital Forecast Database (NDFD). This is a collection of sensible weather elements such as: maximum and minimum temperature, humidity, cloud cover, probability of precipitation, amount of precipitation and wintry precipitation, weather type, and wind direction and speed. In addition to viewing gridded weather data via the internet, more advanced users can decode the individual grids using a "GRIB2 decoder" which can output data as shapefiles, netCDF, GrADS, float files, and comma separated variable files. Specific points in the digital database can be accessed using an XML SOAP service. These capabilities have greatly increased the audience of NDFD data. The NWS has received some criticism from commercial weather vendors for providing graphical forecast data free of charge. They argue that such tailored forecast information compete with their own products. However, a large majority of private weather firms quickly realized its potential benefits and have flourished by using the NDFD as a tool for composing their products.

The same kind of activities are carried out by a number of agencies through the world, in India the same activities are carried out by National Centre for medium range weather forecasting.

7. 8 REMOTE SENSING

Remote sensing is the small- or large-scale acquisition of information of an object or phenomenon, by the use of either recording or real-time sensing device(s) that are wireless, or not in physical or intimate contact with the object (such as by way of aircraft, spacecraft, satellite, buoy, or ship). In practice, remote sensing is the stand-off collection through the use of a variety of devices for gathering information on a given object or area. Thus, Earth observation or weather satellite collection platforms, ocean and atmospheric observing weather buoy platforms, the monitoring of a parolee via an ultrasound identification system, Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), X-radiation (X-RAY) and space probes are all examples of remote sensing. In modern usage, the term generally refers to the use of imaging sensor technologies including: instruments found in aircraft and spacecraft as well as those used in electrophysiology, and is distinct from other imaging-related fields such as medical imaging.

Data processing

Generally speaking, remote sensing works on the principle of the *inverse problem*. While the object or phenomenon of interest (the **state**) may not be directly measured, there exists some other variable that can be detected and measured (the **observation**), which may be related to the object of interest through the use of a data-derived computer model. The common analogy given to describe this is trying to determine the type of animal from its footprints. For example, while it is impossible to directly measure temperatures in the upper atmosphere, it is possible to measure the spectral emissions from a known chemical species (such as carbon dioxide) in that region. The frequency of the emission may then be related to the temperature in that region via various thermodynamic relations.

The quality of remote sensing data consists of its spatial, spectral, radiometric and temporal resolutions.

Spatial resolution

> The size of a pixel that is recorded in a raster image – typically pixels may correspond to square areas ranging in side length from 1 to 1,000 meters (3.3 to 3,300 ft).

Spectral resolution

> The wavelength width of the different frequency bands recorded – usually, this is related to the number of frequency bands recorded by the platform. Current Landsat collection is that of seven bands, including several in the infra-red spectrum, ranging from a spectral resolution of 0.07 to 2.1 μm. The Hyperion sensor on Earth Observing-1 resolves 220 bands from 0.4 to 2.5 μm, with a spectral resolution of 0.10 to 0.11 μm per band.

Radiometric resolution

> The number of different intensities of radiation the sensor is able to distinguish. Typically, this ranges from 8 to 14 bits, corresponding to 256 levels of the gray scale and up to 16,384 intensities or "shades" of colour, in each band. It also depends on the instrument noise.

Temporal resolution

The frequency of flyovers by the satellite or plane, and is only relevant in time-series studies or those requiring an averaged or mosaic image as in deforesting monitoring. This was first used by the intelligence community where repeated coverage revealed changes in infrastructure, the deployment of units or the modification/introduction of equipment. Cloud cover over a given area or object makes it necessary to repeat the collection of said location.

In order to create sensor-based maps, most remote sensing systems expect to extrapolate sensor data in relation to a reference point including distances between known points on the ground. This depends on the type of sensor used. For example, in conventional photographs, distances are accurate in the center of the image, with the distortion of measurements increasing the farther you get from the center. Another factor is that of the platen against which the film is pressed can cause severe errors when photographs are used to measure ground distances. The step in which this problem is resolved is called georeferencing, and involves computer-aided matching up of points in the image (typically 30 or more points per image) which is extrapolated with the use of an established benchmark, "warping" the image to produce accurate spatial data. As of the early 1990s, most satellite images are sold fully georeferenced.

Remote Sensing software

Remote Sensing data is processed and analyzed with computer software, known as a remote sensing application. A large number of proprietary and open source applications exist to process remote sensing data. According to an NOAA Sponsored Research by Global Marketing Insights, Inc. the most used applications among Asian academic groups involved in remote sensing are as follows: ERDAS 36% (ERDAS IMAGINE 25% & ERMapper 11%); ESRI 30%; ITT Visual Information Solutions ENVI 17%; MapInfo 17%. Among Western Academic respondents as follows: ESRI 39%, ERDAS IMAGINE 27%, MapInfo 9%, AutoDesk 7%, ITT Visual Information Solutions ENVI 17%. Other important Remote Sensing Software packages include: TNTmips from MicroImages, PCI

Geomatica made by PCI Geomatics, the leading remote sensing software package in Canada, IDRISI from Clark Labs, Image Analyst from Intergraph, and the original object based image analysis software eCognition from Definiens. Dragon/ips is one of the oldest remote sensing packages still available, and is in some cases free. Open source remote sensing software includes GRASS GIS, QGIS, OSSIM, Opticks (software) and Orfeo toolbox.

7.9 PLANNING MULTILANGUAL APPLICATION

A multilingual application is a system of computer programs which permit a uniform and language independent approach to designing computer applications supporting user interfaces in all the Indian languages. What this means is that a set of software tools is available to users for developing new applications supporting interaction with the computer in one's own mother tongue. The IITM Software is therefore quite different from software such as Editors, Word processors, data bases etc. which may permit data entry and display in Indian Languages. We believe that there are many applications peculiar to India which cannot be handled satisfactorily by adapting existing software which supports input and display in Indian scripts. Text processing with Indian language text requires an approach to dealing with text consistent with the writing systems employed for different languages. An important point in respect of the IIT Madras software is that it includes tools for processing multilingual text. With these tools one would be able to perform very effective string processing on the text and thus cater to applications in linguistics, lexical analysis, parsing etc. The ability to support Roman text along with all the Indian scripts makes the software especially attractive for bilingual applications involving an Indian language and English.

Multilingual Document preparation

This editor program may be readily used to prepare text in all the Indian scripts. Text prepared using the editor may be imported into other applications such as word processors (e.g., Microsoft Word). Also the text may be quickly converted to the html (as well as PostScript, PDF) format for display using standard web browsers. The editor supports a very rich set of aksharas including many not covered by standard coding schemes such as Unicode or ASCII. A point to keep in mind is that almost any desired representation for an akshara can be provided dynamically through externally specified parameters. Printouts of high quality may be produced via postscript or through the word processors into which the text is imported. Data entry is natural and uniform across all Indian languages.

Automatic transliteration across all Indian languages

Applications which require the same text to be displayed in two or more languages simultaneously may be easily handled with the editor. Data entry need be effected for just one script and the text may be reproduced in other scripts automatically. The automatic transliteration feature will be useful for preparing books which deal with the scriptures where knowledge of a particular script may not be required to understand the content.

Generating indexes and concordances for words

The software includes programs for indexing texts so that concordances may be generated for the words in the text. The index generated may also be sorted to yield meaningful word indexes for further study of the manuscripts/text. Seen below is a portion of the concordances generated for words in Tirukkural which consists of 1330 couplets. Each couplet consists of seven words and altogether there are about 6500 words in the text of Kural which are distinct. Shown are the words in sorted order with the number of the "Adhikaram" and the number of the couplet itself. It might appear that the sorting order is not maintained strictly in respect of the word seen on the last line. This is not a bug in the sorting algorithm. The space character in the last line has a code value higher than any Tamil akshara. Hence it falls after the other three words though one would expect it to be placed before the three. When text has to include letters which are not really part of the aksharas, it becomes necessary to treat them differently. In practice, it would be easy enough to handle the space before sorting the string. Likewise, the IITM coding scheme sorts the true consonant (i.e., a pure consonant without a vowel) by placing it after the vowel combinations. This is matter of choice. The algorithm could equally place it at the beginning. Visitors who have learnt Tamil will appreciate this aspect of ordering the aksharas.

Educational aids in Teaching languages and Science

The multilingual capabilities of the system may be effectively used in teaching one language through another. Added to this, the ability to setup web pages makes the system specially attractive to designing computer based training material for use in schools and educational institutions. The link at the right takes you to a sample lesson on Trigonometry (Pythagoras Theorem) in Hindi . The on-line lessons made available at this site for learning Sanskrit stand as excellent examples of educational material prepared using the IITM multilingual software.

Development of Multilingual client applications

Large scale resource sharing across computer systems has been rendered easy on account of the concept of client server applications. The fundamental principle behind a client server application is that the user interface to application is separated from the actual processing of the information. The software, with its library of string processing functions is well suited for developing applications which make use of Indian language user interfaces. Such applications find use with databases, searching though archives of information, on line references etc..

Lesson Number:8 ***Writer: Vishal Verma***

INTRODUTION TO OOPS ***Vetter:***

8.1 OBJECTIVE

The objective of this lesson is to make the students familiar with the basics of object oriented programming language, their features and use of them in programming with examples. After studying this lesson they will be familiar with:

1. Concept of OOPS and their features?
2. Concept of data hiding, encapsulation, operator overloading etc with example?
3. Concept of function overloading, inheritance polymorphism etc?

8.2 INTRODUCTION TO OOPS

Object-oriented programming (OOP) is a programming paradigm that uses "objects" – data structures consisting of data fields and methods together with their interactions – to design applications and computer programs. Programming techniques may include features such as data abstraction, encapsulation, modularity, polymorphism, and inheritance. Many modern programming languages now support OOP. An object-oriented program may be viewed as a collection of interacting *objects*, as opposed to the conventional model, in which a program is seen as a list of tasks (subroutines) to perform. In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an independent 'machine' with a distinct role or responsibility. The actions (or "methods") on these objects are closely associated with the object. For example, OOP data structures tend to 'carry their own operators around with them' (or at least "inherit" them from a similar object or class). In the conventional model, the data and operations on the data don't have a tight, formal association.

Fundamental concepts and features:

- Dynamic dispatch – when a method is invoked on an object, the object itself determines what code gets executed by looking up the method at run time in a table associated with the object. This feature distinguishes an object from an abstract data type (or module), which has a fixed (static) implementation of the operations for all instances. It is a programming methodology that gives modular component development while at the same time being very efficient.

- Encapsulation (or multi-methods, in which case the state is kept separate)
- Subtype polymorphism
- Object inheritance (or delegation)
- Open recursion – a special variable (syntactically it may be a keyword), usually called this or self, that allows a method body to invoke another method body of the same object. This variable is *late-bound*; it allows a method defined in one class to invoke another method that is defined later, in some subclass thereof.

8.3 NEED OF OOPS

The concept of OOPS is needed to overcome some the basic limitations of non object oriented programming languages like : data hiding and polymorphism. Though the programmer can build effective programmers by using procedural languages yet the basic problems cannot be overcome without the introduction of the concept of OOPS. The object oriented programming elements like class and objects provide that basic feature to the programmer by using which the programmer can build the highly secure and effective programs. Another feature like encapsulation is so useful, that by using it the programmer can keep the data and the related operations together under one roof, i.e can be bound within same boundary. This feature of combining together the data and function helps by the small range of domain in which the application is to be used. Encapsulation ultimately helps the programmer to think over the requirements which are supposed to be fulfilled over the specified range only.

8.4 CLASSES AND OBJECTS

A **class** is a construct that is used as a blueprint (or template) to create objects of that class. This blueprint describes the state and behavior that the objects of the class all share. An object of a given class is called an instance of the class. The class that contains (and was used to create) that instance can be considered as the type of that object, e.g. an object instance of the "Fruit" class would be of the type "Fruit". a class is a cohesive package that consists of a particular kind of metadata. A class has both an interface and a structure. The interface describes how to interact with the class and its instances using methods, while the structure describes how the data is partitioned into attributes within an instance. A class may also have a representation (metaobject) at run time, which provides run time support for manipulating the class-related metadata. In object-oriented design, a class is the most specific type of an object in relation to a specific layer.

Why Classes?

Classes can accelerate development by reducing redundant program code, testing and bug fixing. If a class has been thoroughly tested and is known to be a 'solid work', it is usually true that using or extending the well-tested class will reduce the number of bugs - as compared to the use of freshly-developed or ad hoc code - in the final output. In addition, efficient class reuse means that many bugs need to be fixed in only one place when problems are discovered.

Another reason for using classes is to simplify the relationships of interrelated data. Rather than writing code to repeatedly call a graphical user interface (GUI) window drawing subroutine on the terminal screen (as would be typical for structured programming), it is more intuitive. With classes, GUI items that are similar to windows (such as dialog boxes) can simply inherit most of their functionality and data structures from the window class. The programmer then need only add code to the dialog class that is unique to its operation. Indeed, GUIs are a very common and useful application of classes, and GUI programming is generally much easier with a good class framework.

Structure of a class

Along with having an interface, a class contains a description of structure of data stored in the instances of the class. The data is partitioned into attributes (or *properties*, *fields*, *data members*). Going back to the television set example, the myriad attributes, such as size and whether it supports color, together comprise its structure. A class represents the full description of a television, including its attributes (structure) and buttons (interface).

Some languages allow an implementation of a class to specify constructor (or *initializer*) and destuctor (or *finalizer*) methods that specify how instances of the class are created and destroyed, respectively. A constructor that takes arguments can be used to create an instance from passed-in data. The main purpose of a constructor is to establish the invariant of the class, failing if the invariant isn't valid. The main purpose of a destructor is to destroy the identity of the instance, invalidating any references in the process. Constructors and destructors are often used to reserve and release, respectively, resources associated with the object. In some languages, a destructor can return a value which can then be used to obtain a public representation (transfer encoding) of an instance of a class and simultaneously destroy the copy of the instance stored in current thread's memory.

Information hiding and encapsulation

Many languages support the concept of information hiding and encapsulation, typically with access specifiers for class members. Access specifiers specify constraints on who can access which class members. Some access specifiers may also control how classes inherit such constraints. Their primary purpose is to separate the interface of a class from its implementation.

A common set of access specifiers that many object-oriented languages support is:

- **Private** (or *class-private*) restricts the access to the class itself. Only methods that are part of the same class can access private members.

- **Protected** (or *class-protected*) allows the class itself and all its subclasses to access the member.
- **Public** means that any code can access the member by its name.

Categories of classes

Concrete classes

A concrete class is a class that can be instantiated. This contrasts with abstract classes as described below.

Abstract class

An abstract class, or *abstract base class* (ABC), is a class that cannot be instantiated. Such a class is only meaningful if the language supports inheritance. An abstract class is designed *only* as a parent class from which child classes may be derived. Abstract classes are often used to represent abstract concepts or entities. The incomplete features of the abstract class are then shared by a group of subclasses which add different variations of the missing pieces.

Abstract classes are superclasses which contain abstract methods and are defined such that concrete subclasses are to extend them by implementing the methods. The behaviors defined by such a class are "generic" and much of the class will be undefined and unimplemented. Before a class derived from an abstract class can become concrete, i.e. a class that can be instantiated, it must implement particular methods for all the abstract methods of its parent classes.

Sealed classes

Some languages also support sealed classes. A sealed class cannot be used as a base class. For this reason, it also cannot be an abstract class. Sealed classes are primarily used to prevent derivation. They add another level of strictness during compile-time, improve memory usage, and trigger certain optimizations that improve run-time efficiency.

Inner class

One common type is an inner class, which is a class defined within another class. Since it involves two classes, this can also be treated as another type of class association. The methods of an inner class can access static methods of the enclosing class(es). An inner class is typically not associated with instances of the enclosing class, i.e. an inner class is not instantiated along with its enclosing class. Depending on language, it may or may not be possible to refer to the class from outside the enclosing class. A related concept is *inner types* (a.k.a. *inner data type*, *nested type*), which is a generalization of the concept of inner classes. C++ is an example of a language that supports both inner classes and inner types (via *typedef* declarations).

Named vs. anonymous classes

In most languages, a class is bound to a name or identifier upon definition. However, some languages allow classes to be defined without names. Such a class is called an anonymous class (analogous to named vs. anonymous functions).

Metaclasse

Metaclasses are classes whose instances are classes. A metaclass describes a common structure of a collection of classes. A metaclass can implement a design pattern or describe a shorthand for particular kinds of classes. Metaclasses are often used to describe frameworks.

In some languages such as Pythron, Ruby, Java, and Smalltalk, a class is also an object; thus each class is an instance of the unique metaclass, which is built in the language. For example, in Objective-C, each object and class is an instance of NSObject. The Common Lisp Object System (CLOS) provides metaobject protocols (MOPs) to implement those classes and metaclasses.

Partial classes

Partial classes are classes that can be split over multiple definitions (typically over multiple files), making it easier to deal with large quantities of code. At compile time the partial classes are grouped together, thus logically make no difference to the output. An example of the use of partial classes may be the separation of user interface logic and processing logic. A primary benefit of partial classes is allowing different programmers to work on different parts of the same class at the same time. They also make automatically generated code easier to interpret, as it is separated from other code into a partial class.

Objects

In the "real" world, objects are the entities of which the world is comprised. Everything that happens in the world is related to the interactions between the objects in the world. Just as atoms, which are objects, combine to form molecules and larger objects, the interacting entities in the world can be thought of as interactions between and among both singular ("atomic") as well as compound ("composed") objects. The real world consists of many, many objects interacting in many ways. While each object may not be overly complex, their myriad of interactions creates the overall complexity of the natural world. It is this complexity that we wish to capture in our software systems. *The existence of data in an object is an implementation technique used to generate the required behavior of that object.*

In an object-oriented software system,*objects* are entities used to represent or model a particular piece of the system.

Objects are the primary units used to create abstract models.

There are a number of schools of object-oriented programming, which differ slightly on how they view objects. Here, we will take a "behaviorist" (our term) stance:

An object is characterized solely by it behaviors.

Essentially this defines an object by the way it interacts with its world. An object that does not interact with anything else effectively does not exist. Access to internally stored data is necessarily through some sort of defined behavior of the object. It is impossible for an outside entity to truly "know" whether or not a particular piece of data is being stored inside of another object.

8.5 DATA HIDING

Data hiding is the method of writing the code in such a way that the actual code will remain hidden from the other persons who uses that data. The purpose of hiding the data is to keep the basic details hidden from the other and hence to keep our data secure and safe. A number of methods are used to hide the data, the OOP languages like C++ support this concept through encapsulation (discussed in next section). The programmer through access specifier like public, private etc can control the visibility of the data which is to be shown and which is to be hide. The general trend is to hide the data members and to show the member function is followed by programmer in the object oriented programming languages. After writing code to hide the data, the user is able to use only the publically specified member functions.

8.6 DATA ENCAPSULATION

Encapsulation as "the process of compartmentalizing the elements of an abstraction that constitute its structure and behavior; encapsulation serves to separate the contractual interface of an abstraction and its implementation." This formulation is cited by a number of books as an authoritative definition of encapsulation.

The purpose is to achieve potential for change: the internal mechanisms of the component can be improved without impact on other components, or the component can be replaced with a different one that supports the same public interface. Encapsulation also protects the integrity of the component, by preventing users from setting the internal data of the component into an invalid or inconsistent state. Another benefit of encapsulation is that it reduces system complexity and thus increases robustness, by limiting the interdependencies between software components.

Encapsulating software behind an interface allows the construction of objects that mimic the behavior and interactions of objects in the real world. For example, a simple digital alarm clock is a real-world object that a lay person can use and understand. They can understand what the alarm clock does, and how to use it through the provided interface (buttons and screen), without having to understand every part inside of the clock. Similarly, if you replaced the clock with a different model, the lay person could continue to use it in the same way, provided that the interface works the same.

In the more concrete setting of an object-oriented programming language, the notion is used to mean either information hiding mechanism, a bundling mechanism, or the combination of the two.

```
class Exforsys
{
public:
int sample();
int example(char *se)
int endfunc();
.........
......... //Other member functions

private:
int x;
float sq;
..........
......... //Other data members
};
```

in the example discussed above all the data members declared within the specifier public are available outside the boundary of the class and the other which are declared within the specifier private are available directly for the use by the user but they can be accessed only through public member function of the class.

8.7 OPERATOR OVERLOADING

Operator overloading is a specific case of polymorphism in which some or all of operators like +, =, or == have different implementations depending on the types of their arguments. Sometimes the overloading is defined by the language; sometimes the programmer can implement support for new types.

Operator overloading is claimed to be useful because it allows the developer to program using notation "closer to the target domain" and allows user-defined types a similar level of syntactic support as types built into the language. It can easily be emulated using function calls; for an example, consider the integers a, b, c:

a + b * c

In a language that supports operator overloading, and assuming the '*' operator has higher precedence than '+', this is effectively a more concise way of writing:

add (a, multiply (b,c))

The operator keyword declares a function specifying what operator-symbol means when applied to instances of a class. This gives the operator more than one meaning, or "overloads" it. The compiler distinguishes between the different meanings of an operator by examining the types of its operands.

type operator operator-symbol (parameter-list)

The name of an overloaded operator is operatorx, where x is the operator. For example, to overload the addition operator, you define a function called operator+. Similarly, to overload the addition/assignment operator, +=, define a function called operator+=.

Example :

```
Time operator+(const Time& lhs, const Time& rhs) {
    Time temp = lhs;
    temp.seconds += rhs.seconds;
    if (temp.seconds >= 60) {
        temp.seconds -= 60;
        temp.minutes++;
        }
    temp.minutes += rhs.minutes;
    if (temp.minutes >= 60) {
        temp.minutes -= 60;
        temp.hours++;
        }
    temp.hours += rhs.hours;
    return temp;
    }

Time Time::operator+(const Time& rhs) const {
    Time temp = *this;  /* Copy 'this' which is not to be modified */
    temp.seconds += rhs.seconds;
    if (temp.seconds >= 60) {
        temp.seconds -= 60;
        temp.minutes++;
        }
    temp.minutes += rhs.minutes;
    if (temp.minutes >= 60) {
        temp.minutes -= 60;
        temp.hours++;
        }
    temp.hours += rhs.hours;
    return temp;
    }
```

The operators in the table below shows the operator which cannot be overloaded.

Operator	Name
.	Member selection

.*	Pointer-to-member selection
::	Scope resolution
? :	Conditional
#	Preprocessor convert to string
##	Preprocessor concatenate

8.8 FUNCTION OVERLOADING

Function overloading or method overloading is a feature found in various programming languages such as C#, VB.Net, C++, D and Java that allows the creation of several methods with the same name which differ from each other in terms of the type of the input and the type of the output of the function. For example, *doTask()* and *doTask(object O)* are overloaded methods. To call the latter, an object must be passed as a parameter, whereas the former does not require a parameter, and is called with an empty parameter field. A common error would be to assign a default value to the object in the second method, which would result in an *ambiguous call* error, as the compiler wouldn't know which of the two methods to use.

Another example would be a *Print(object O)* method. In this case one might like the method to be different when printing, for example, text or pictures. The two different methods may be overloaded as *Print(text_object T); Print(image_object P)*. If we write the overloaded print methods for all objects our program will "print", we never have to worry about the type of the object, and the correct function call again, the call is always: *Print(something)*.

Method overloading is usually associated with statically-typed programming languages which enforce type checking in function calls. When overloading a method, you are really just making a number of different methods that happen to have the same name. It is resolved at compile time which of these methods is used. Method overloading should not be confused with virtual functions, where the correct method is chosen at runtime.

```
Bill()
 { tip = 15.0, total = 0.0 }
Bill(double setTip, double setTotal)
 { tip = setTip, total = setTotal }

Bill cafe(10.00, 4.00);
```

8.9 INHERITANCE AND POLYMORPHISM

In object-oriented programming (OOP), Inheritance is a way to compartmentalize and reuse code by creating collections of attributes and behaviors called objects which can be based on previously created objects. In *classical inheritance* where objects are defined by classes, classes can inherit other classes. The new classes, known as Sub-classes (or derived classes), inherit attributes and behavior of the pre-existing classes, which are referred to as Super-classes (or ancestor classes). The inheritance relationship of sub- and super-classes gives rise to a hierarchy. In Prototype-based programming objects can be defined directly from other objects without the need to define any classes.

Inheritance should not be confused with (subtype) polymorphism, commonly called just polymorphism. Inheritance is a relationship between implementations, whereas subtype polymorphism is relationship between types (interfaces in OOP). (Compare connotation/denotation.) In some, but not all OOP languages, the notions coincide because the only way to declare a subtype is to define a new class that inherits the implementation of another.

Inheritance does not entail behavioral sub-typing either. It is entirely possible to derive a class whose object will behave incorrectly when used in a context where the parent class is expected; see the Liskov substitution principle.

Polymorphism

Polymorphism in the context of object-oriented programming, is the ability of one type, A, to appear as and be used like another type, B. The purpose of polymorphism is to implement a style of programming called *message-passing* in the literature, in which objects of various types define a common interface of operations for users.

In strongly typed languages, polymorphism usually means that type A somehow derives from type B, or type C implements an interface that represents type B. In weakly typed languages types are implicitly polymorphic.

The primary usage of polymorphism in industry (object-oriented programming theory) is the ability of objects belonging to different types to respond to method, field, or property calls of the same name, each one according to an appropriate type-specific behavior. The programmer (and the program) does not have to know the exact type of the object in advance, and so the exact behavior is determined at run-time (this is called *late binding* or *dynamic binding*).

The different objects involved only need to present a compatible interface to the clients' (calling routines). That is, there must be public or internal methods, fields, events, and properties with the same name and the same parameter sets in all the super-classes, subclasses and interfaces. In principle, the object types may be unrelated, but since they share a common interface, they are often implemented as subclasses of the same super-class. Though it is not required, it is understood that the different methods will also produce similar results (for example, returning values of the same type).

Polymorphism is not the same as method overloading or method overriding. Polymorphism is only concerned with the application of specific implementations to an interface or a more generic base class. Method overloading refers to methods that have the same name but different signatures inside the same class. Method overriding is where a subclass replaces the implementation of one or more of its parent's methods. Neither method overloading nor method overriding are by themselves implementations of polymorphism.

Example :

```
#include <iostream>
#include <string>

using namespace std;

class Animal
{
        public:
        Animal(const string& name) : name(name) {}
        virtual string talk() = 0;
        const string name;
};

class Cat : public Animal
{
        public:
        Cat(const string& name) : Animal(name) {}
        virtual string talk() { return "Meow!"; }
};

class Dog : public Animal
{
        public:
        Dog(const string& name) : Animal(name) {}
        virtual string talk() { return "Arf! Arf!"; }
};
```

```
// prints the following:
//
// Missy: Meow!
// Mr. Mistoffelees: Meow!
// Lassie: Arf! Arf!
//
int main()
{
        Animal* animals[] =
        {
                new Cat("Missy"),
                new Cat("Mr. Mistoffelees"),
                new Dog("Lassie")
        };

        for(int i = 0; i < 3; i++)
        {
                cout << animals[i]->name << ": " << animals[i]->talk() << endl;
                delete animals[i];
        }
        return 0;
}
```