

**PC SOFTWARE AND
PROGRAMMING IN C
(DSCSC11)
(BSC COMPUTER SCIENCE-I)**



ACHARYA NAGARJUNA UNIVERSITY

CENTRE FOR DISTANCE EDUCATION

NAGARJUNA NAGAR,

GUNTUR

ANDHRA PRADESH

Lesson 1:**BASIC CONCEPTS****1.0 Objectives:**

The objective of this lesson is primarily to define what a computer is and to give a brief insight into the evolution of computer, its types and the use of computer in our regular life. After the completion of this lesson the student will be able to understand:

- What a computer is.
- Characteristics and limitations of a Computer.
- History of Computers.
- The evolution of Computer through five generations.
- Types of Computers.
- Advantages of using Computers.
- Applications of Computers.

Structure of the Lesson:

- 1.1. Introduction
- 1.2. What is a Computer?
- 1.3. Characteristics of a computer
- 1.4. Limitations of a computer
- 1.5. Major components of a computer
- 1.6. History of Computers
- 1.7. Computer Generations
- 1.8. Types of Computers
- 1.9. Advantages of Computer
- 1.10. Applications of Computers
- 1.11. Summary
- 1.12. Technical Terms
- 1.13. Model Questions
- 1.14. References

1. 1. Introduction:

We are living in an information age. Every type of job relies upon getting information, using it, managing it, and relaying information to others. Computers enable the efficient processing and storage of information. The information processed by the computer becomes digital. So Computers are directly responsible for the smooth running of this information age. Computers are placed everywhere we look, and even in places we can't see. Computers check out our groceries, book tickets for our journey, fill fuel in our vehicles, dispense money at the ATM, turn the heat on and off, and control the way our car runs. They're everywhere!

The word "Computer" is derived from the word "COMPUTE", which means to calculate.

Before learning more about computers it is suggested that one should have a basic idea of what data, information and instruction is.

Data:

Facts, symbols, names, numbers or anything, which has a little meaning, is called as **Data**. **Ex:** hanuman, 123, ramu, 10:50 etc.

Information:

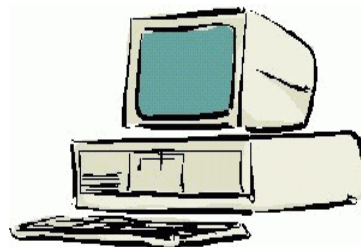
Processed data is known as information.

Instruction:

Instruction is a command, which operates on the stored data.

1.2.What is a Computer?

Computer is an electronic device, which takes data as input, processes it and produces information as output.



1.3. Characteristics of a Computer:

Computers are becoming popular day by day because of their continuously increasing power and its usefulness. The following are some of its characteristics.

Accuracy:

Computers always produce correct calculations. If mistake occurs in any calculation, they are due to human errors like entering wrong data or incorrect instruction.

Speed:

Computers can carry out instructions in less than one millionth of a second.

Versatility:

Computer can perform any type of work.

Storage capacity:

Computers have the capability to store large amounts of instructions and data. When we ask for data it supplies the stored information to us.

Repetitiveness:

Computers are capable of performing any task given to them repeatedly.

1.4. Limitations of Computers:**Lack of Commonsense:**

Computer is only a tool. It cannot think. It does not have commonsense or intelligence of its own. Computer acts like a human upto some extent.

Inability to correct:

A computer cannot correct wrong instructions by itself.

Dependence on human instructions:

A computer cannot generate any information on it's own. We must give the correct instructions.

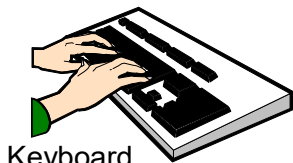
1.5. Major components of a computer:

A System is defined as a set of interacting elements, responding to inputs so as to produce outputs. The computer system consists of the following elements:

Hardware:

The physical components of a computer system are called as hardware. i.e. components, which can be seen and touched. It includes input devices, output devices, electronic circuits, capacitors etc.

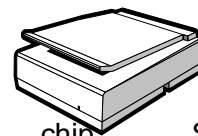
The following are some of the hardware components.



Keyboard



Floppy Disk



chip

Scanner

Software:

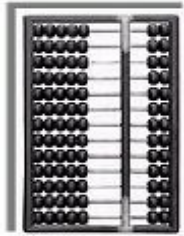
A program or a collection of programs is known as software. Program is a sequence of instructions designed to direct a computer to perform certain tasks.

Note: A floppy disk is a hardware, but contains a program stored on it, which is a software.

1.6. History of Computers:

History of computer could be traced back to the effort of man to count large numbers. Computer history starts with the development of a device called ABACUS by Chinese around 3000 BC. This device was used for counting.

Calculating Machines:



It took over generations for early man to build mechanical devices for counting large numbers. The Egyptian and Chinese people developed the first calculating device called ABACUS. The word ABACUS means calculating board. It has a number of horizontal bars each having ten beads. Horizontal bars represent units, tens, hundreds, etc.

Napier's Bones:

In the early 1600's, a Scottish mathematician called **John Napier** invented a tool called Napier's Bones, which were multiplication tables inscribed on strips of wood or bone. He also invented *logarithms*, which is a technology that allows multiplication to be performed via addition.

Slide Rule:

English mathematician **Edmund Gunter** developed the slide rule. This machine could perform operations like addition, subtraction, multiplication and division. It was widely used in Europe in 16th century.

Pascal's Adding and Subtracting Machine:

The mechanical adding machine was invented by a 19 year old French boy named **Blaise Pascal** way back in the year 1642. This machine could perform addition and subtraction. The machine consisted of wheels, gears and cylinders.

Leibniz's Multiplication and Dividing Machine:

Blaise Pascal machine was further improved by **Gottfried Wilhelm von Leibniz**, a German mathematician to perform all four arithmetic operations. This machine could perform both multiplication and division.

Babbage's Analytical Engine:

It was in the year 1823, a famous English man **Charles Babbage** designed a mechanical computing machine to do complex mathematical calculations. It was called *difference engine*. Later he developed a general-purpose calculating machine called *analytical engine*. Charles Babbage is called the *father of computer*.



Augusta Ada Byron –Countess of Lovelace:

In the year **1842**, Ada Lovelace with her set of instructions became the pioneer of modern programming languages and historians have accorded her the status of world's first computer programmer.

Mechanical and Electrical Calculator:

In the beginning of 19th century the mechanical calculator was developed to perform all sorts of mathematical calculations. It was widely upto 1960s. Later the rotating part of mechanical calculator was replaced by electric motor. So it was called the electrical calculator.

Modern Electronic Calculator:

The electronic calculator used in 1960s was run with vacuum tubes, which was quite bulky. Later it was replaced with *transistors* and as a result the size of calculators became too small. The modern electronic calculator can compute all kinds of mathematical computations and mathematical functions. It can also be used to store some data permanently. Some calculators have in-built programs to perform some complicated calculations.

1.7. Computer Generations:

The evolution of computer started from 16th century and resulted in the form that we see today. This period, during which the evolution of computer took place, can be divided into five distinct phases known as *Generations of Computers*. Each phase is distinguished from others on the basis of the type of technology used.

First Generation (Period 1951- 58):

First generation computers used *Thermion valves or Vacuum tubes* for circuitry, magnetic drums for memory, and were often enormous, occupying entire room. They were very expensive to operate and in addition to using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions. First generation computers relied on machine language to perform operations and could only solve one problem at a time. Input was based on punched cards and paper tape, and output was printed on paper (printouts).

Some of the computers of this generation were: UNIVAC, ENIAC, EDVAC, EDSAC.

Second Generation (Period 1959-64):

With the invention of the Transistor at Bell labs in 1947 by William Shockley, John Bardeen and Walter Brattain, it became clear that transistors would replace vacuum tubes. Transistors are smaller than vacuum tubes and have higher operating speed, more reliable, used less energy and generated less heat. Thus the size of the computer got reduced considerably. The transistor was invited into Second-generation, computers moved from binary machine language to symbolic or assembly language. This allowed programmers to specify instructions in words. High-level programming languages were also being developed at this time. These were also the first computers that stored their instructions in their memory, which moved from a magnetic drum to magnetic core technology. It is in the second generation that the concept of Central Processing Unit (CPU), memory, programming language, input and output units were developed. The programming languages such

as COBOL, FORTRAN were developed during this period. Some of the computers of the Second Generation were IBM 1620, IBM 1401, CDC 3600.

Third Generation (1965 to 1970):

The third generation computers were introduced in 1964. They used *Integrated Circuits* (IC's). These IC's are popularly known as *Chips*. Transistors were miniaturized and placed on silicon chips, called semiconductors, which drastically increased the speed and efficiency of computers. A single IC has many transistors, registers and capacitors built on a single thin slice of silicon. So it is quite obvious that the size of the computer got further reduced. Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors interfaced with an operating system. Operating system allowed the device to run many different applications at one time with a central program that monitored the memory. Computers of these generations were small in size, low cost, large memory and processing speed is very high. High-level language such as BASIC (Beginners All purpose Symbolic Instruction Code) was developed during this period.

Some of the computers developed during this period were IBM-360, ICL-1900, IBM-370, and VAX-750.

Fourth Generation (1971 to present):

With the invention of the microprocessor, thousands of integrated circuits were built onto a single silicon chip. The present day computers that we see today are the fourth generation computers that started around 1975. It uses *large scale Integrated Circuits* (LSIC) built on a single silicon chip called microprocessor. Due to the development of microprocessor it is possible to place computer's *central processing unit* (CPU) on single chip. These computers are called microcomputers. Later *very large scale Integrated Circuits* (VLSIC) replaced LSICs.

In 1981 IBM introduced its first personal computer for the home user, and in 1984 Apple introduced the Macintosh. Microprocessors moved out of the realm of desktop computers into many areas of life as more and more everyday products began to use microprocessors.

As these small computers became more powerful, they were linked together to form networks, which eventually led to the development of the Internet. Fourth generation computers also saw the development of GUIs(Graphical User Interface), the mouse and handheld devices.

Fifth Generation:

Fifth generation computing devices, based on artificial intelligence, are still in development, though there are some applications, such as voice recognition, that are being used today. The speed is extremely high in fifth generation computer. The use of parallel processing and superconductors is helping to make artificial intelligence a reality. Quantum computation, molecular and nanotechnology will radically change the face of computers in years to come. The goal of fifth generation computing is to develop devices that respond to natural language input, capability of learning by themselves and also being self-organized.

First Generation:

- ❑ Vacuum tubes for internal operations
- ❑ Primary Memory Limited
- ❑ Heat and Maintenance problem
- ❑ Punched cards for input and output

Second Generation:

- ❑ Transistors for internal operations
- ❑ Smaller than vacuum-tube-built computers
- ❑ Thousand times faster than first-generation computers
- ❑ Magnetic tapes and disks for secondary storage
- ❑ Increased processing speed and reliability
- ❑ Increased use of high-level languages

Third Generation:

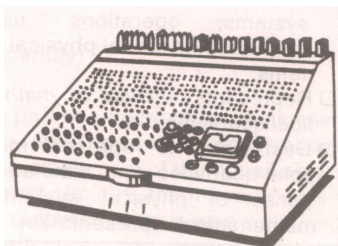
- ❑ Hundreds of transistors packed into a single integrated circuit on a silicon chip for internal operations
- ❑ Dramatic reduction in size and cost
- ❑ Significant increase in reliability, speed, and efficiency
- ❑ Increased primary memory capacity

Fourth Generation:

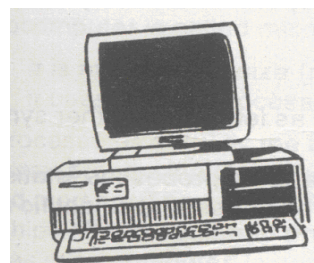
- ❑ Large scale and VLSI for internal operations
- ❑ Development of the microprocessor
- ❑ Greater versatility in software's
- ❑ Introduction of microcomputers and super computers

1.8. Types of Computers:

Computers can be classified into two categories: Analog and Digital computers



Analog Computer



Digital Computer

Analog computers operate by measuring rather than counting. These computers use signals as input, which can come from devices like speedometers, thermometers etc. These signals are continuous flow of physical quantities like speed, temperature, pressure etc. Analog computer has only limited memory facility and is restricted in the type of calculations it can perform.

Digital computers represents data in the form digits and all operations are done on these digits at extremely fast rates. Digital computers basically know how to count the digits. These computers are mostly used in engineering and scientific calculations. Digital watches are good example of a digital computer.

Hybrid computer is a combination of both analog and digital computers. i.e., a part of processing is done on analog and a part on digital computer. It provides greater precision than can be attained with analog computers, greater control and speed than is possible with digital computers. It can accept input data in both analog and digital form. It is used for simulation applications.

Digital Computers are divided into the following categories:

- Mini Computers
- Mainframe computers
- Micro Computers
- Super Computers
-

Now let us discuss the varieties of digital computers that we see today. They can be divided into different categories depending upon the size, efficiency, memory and number of users.

Microcomputer:

Microcomputer is at the lowest end of the computer range in terms of speed and storage capacity. These are most widely used computers which we commonly refer to as “Personal Computers” or “Desktop Computer”. Its CPU is a microprocessor. The PC supports a number of input and output devices. It can be used by only one user at a time. Examples of microcomputer are IBM PC, PC-AT.

Mini Computer:

Mini computer is designed to support more than one user at a time. It possesses large storage capacity and operates at a higher speed. The mini computer is used in a multi-user environment in which various users can work at the same time. This type of computer is generally used for processing large volume of data in an organization. They are also used as servers in Local Area Networks (LAN).

Mainframes:

Mainframe computer is a computer, which has capability to support many terminals and peripherals. These computers operate at very high speed, have very large storage capacity and can handle the workload of many users. They are generally used in management of centralized databases. They are also used as controlling nodes in Wide Area Networks (WAN). Examples of mainframes are ICL, IBM 3000 series and DEC.

Super computer:

Super computers are the fastest and most expensive machines. They have high Processing speed compared to other computers. They also have multiprocessing techniques. These computers are used widely in complex scientific applications like weather forecasting, genetic engineering, biomedical research, remote sensing, aircraft design etc. One of the ways in which super computers are built is by interconnecting hundreds of microprocessors. Examples of supercomputers are CRAY YMP, CRAY2, CRAY XMP, PARAM from India and IBM's Bluegear.

1.9. Advantages of Computers:

- High rate of processing speed with reliability.
- Huge volumes of data can be stored in the memory and any particular data item can be retrieved at random in a fraction of a second.
- Solution of a much-complicated problem is feasible at a greater speed.
- Processing of voluminous data saves a good lot of clerical labor, which in turn reduces the processing cost.
- Performs operations efficiently even in environments inaccessible to humans, such as furnace, poisonous atmosphere, vacuum, unmanned satellite etc.

1.10. Applications of Computers:

Computers were initially used for prime applications in the field of mathematics and science. Their first effective utilization was made in the field of military for decoding the messages of the enemy. Then they entered the application of real time control systems like landing on moon. Currently maximum number of computers are being used in business and commercial sectors. The following are the list of applications:

- Scientific applications
- Weather forecasting
- Online reservations
- Business and office
- Banking sector
- Electronics funds transfer (EFT)
- Electronic commerce
- Process control
- Computer aided design (CAD)
- Computer aided manufacturing (CAM)
- Electronic mail (e-mail)

1.11. Summary:

In this lesson we discussed the major characteristics of a computer. The speed, accuracy, memory and versatility are some of the features associated with a computer. Computer performs operations automatically. Computer system involves two major components hardware and software. Hardware refers to physical components of a system and software refers to a set of programs. The computer that we see today was not developed over night. It has taken centuries of human effort to see the computer in its present form today. Computers history began with the development of a counting device called ABACUS, later several evaluations including analog machines, mechanical calculators, electrical machine and other modern computers. Computers are classified into several generations based on the technology used in building the computer. There are five generations of computers. Over these generations the physical size of computer has decreased, and on the other hand the processing speed of computer has improved tremendously. The Computers have proved not only their presence but also their effectiveness in whatever field they foray.

1.12. Technical Terms:

Chip: A tiny wafer of silicon containing miniature electric circuits that can store million of bits of information.

VLSI (Very Large Scale Integration): The use of integrated circuits with 100,000 to a million logic gates.

Artificial intelligence (AI): The study of thinking process of humans and representation of those processes via machines (computer, robots, etc.)

Microprocessor :A complete central processing unit (CPU) contained on a single silicon chip.

Laptop: Small, lightweight, portable battery-powered computers that can fit onto our lap. They have a thin, flat, liquid crystal display screen.

E-mail (Electronic mail): A service that sends messages on computers via local or global networks.

1.13. Model Questions:

1. What is a computer and explain the characteristics of a computer?
2. What are the major limitations of a computer?
3. What is Software and Hardware?
4. Briefly explain History of computers?
5. Explain Generations of computers?
6. Write about different types of computers?
7. Write short notes on advantages and applications of computers?

1.14.References:

Prof.G.S.Prasad, M.Sc, M.S “**Fundamentals of Computer - Telugu Academy**”

Dharma Raja Raman “**Computer Primer**”

Alexis Leon & Mathews Leon “**Fundamentals of Information Technology**”

A.Jaiswal “**Fundamentals of computers and Information technology**”

N.Subramainian “**Computer Genesis, Programming and applications**”

**G.RAMANJIAH .,M.Sc (Computers)
Lecturer, Dept. of Computer Science,
JKC College, GUNTUR.**

Lesson: 2**COMPONENTS OF A COMPUTER****2.0 Objectives:**

The main objectives of this lesson are:

- To understand the four basic operations of the computer.
- To know the basic components of the computer and their functions.
- To know the function of memory and different kinds of memory.
- To understand the concept of cache memory and registers.
- To know about input and output devices.

Structure of the lesson:**2.1. Introduction****2.2. Operations of a computer system****2.3. Block diagram of a computer****2.4. Types of memory****2.4.1. Primary memory****2.4.2. Secondary memory****2.5. Cache memory****2.6. Registers****2.7. Input devices****2.8. Output devices****2.9. Summary****2.10. Technical terms****2.11. Model Questions****2.12. References****2.1. Introduction**

In the previous lesson we discussed about the history and evolution of computer. In this lesson we will have an overview of the basic design of a computer. We will know how different parts of a computer are organized and how various operations are performed between different parts to do a specific task. The internal architecture of computer may differ from system to system, but the basic organization remains the same for all computer systems.

2.2. Operations of a Computer System:

Computer system has four main stages: The **Input stage**- where data is put into the computer. The **Processing stage** - where the data is processed. **Storage** and The **Output stage** - where the results are displayed.

Input:

Input is supplied to the computer with the use of a keyboard, a mouse, or another input device. These input devices may be called **peripheral devices**.

Processing:

Processing is done inside the computer in an area called the **central processing unit (CPU)**. Processing is the conversion of input to output.

Storage:

Storage refers to holding information in a specific place.

Output:

Output is the result of a computer process. Output may be viewed on a monitor screen, heard through speakers, printed on paper with printers, and so forth.

2.3. Block diagram of a computer:

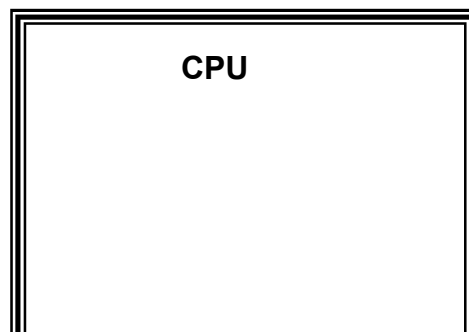
Basic functional units of a digital computer includes three major parts:

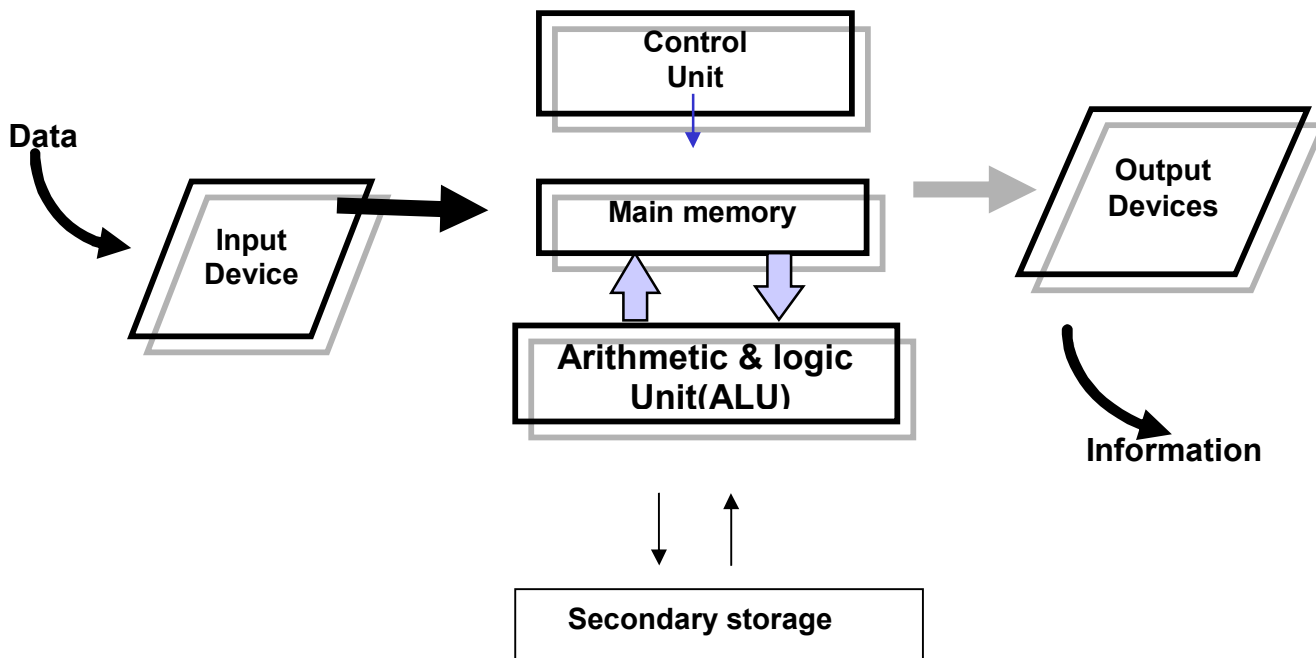
- Input unit
- Central processing unit (CPU)
- Output unit

Input Unit:

Computer processes data to produce some form of useful information as output. To do this, a computer needs some kind of data (facts, numbers, characters etc) to be processed called **Input**. So we need a device to enter data into the computer. Input unit writes the instructions and data into a computer's memory. Input unit of computer is similar to our 'ears' and 'eyes' through which we receive data. Input devices deal with original data. Various input devices in use are: Keyboard, Mouse, Scanner, light pen, Graphic Tablet, Touch panel, Optical Mark reader, Magnetic tapes, Magnetic drums.

Block Diagram of a Computer



**Central Processing Unit (CPU):**

The central processing unit (CPU) is an essential component of a computer. Because it is the part that executes the programs and controls the operations of all the hardware. The central processing unit (CPU) is like our brain.

It consists of three units:

1. Arithmetic and Logic unit (ALU)
2. Control unit (CU)
3. Memory unit (MU)

ALU (Arithmetic and Logic Unit):

Arithmetic and logic unit is the part of the CPU, where arithmetic and logical functions are performed, i.e., mathematical or arithmetic functions like addition, subtraction, multiplication, division etc. Logical operations like logical AND, OR, NOT.

Control Unit: The Control unit is the circuitry that controls the flow of information through the processor and co-ordinates the activities of the other units within it. It also controls transfer of data between CPU and memory and CPU and I/O devices.

Memory Unit: Memory stores the input from input device and output from the CPU, as well as the instructions that are followed by the CPU. All the instructions, data, intermediate results of any calculation and final output in the memory can be retrieved when necessary. The data and commands are stored in the addresses called memory locations.

The amount of storage in the memory can be measured. The units of measurement of memory are

bit=0 or 1
 nibble= 4 bits
 1byte=8 bits
 1 Kilo byte (KB)= 1024 bytes
 1 Mega byte (MB)=1024 KB
 1 Giga byte (GB) = 1024 MB

There are two types of memory, primary or main memory and secondary or auxiliary memory.

Output Unit: Output devices allow us to get information from the computer. Various output devices are: Monitor, Printer, Speakers, CRT – Cathode Ray Tube, Plasma, LCD – Liquid Crystal Display.

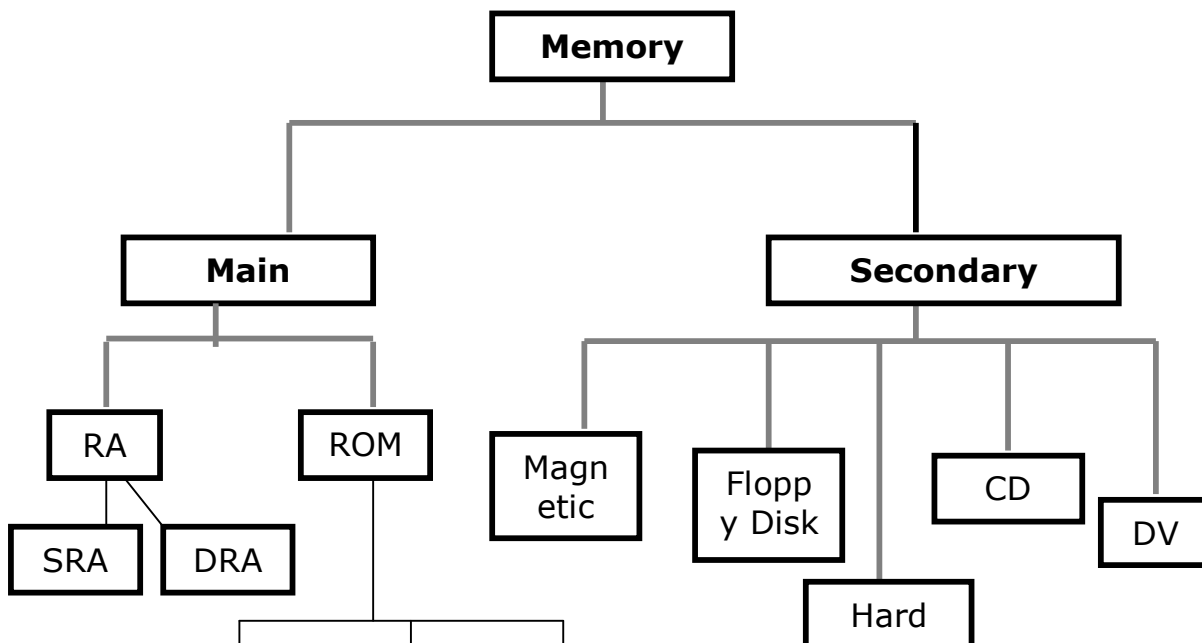
2.4. Types of Memory:

Memory:

Memory is the storage area where the computer stores the instructions for solving a problem, the data to be processed, the intermediate results and the final results until they are displayed.

Types of Memories: Memory can be divided into two main categories:

- Main Memory or Primary Memory
- Secondary Memory or Auxiliary memory



Primary memory consists of data that are actively being used. It is typically high-speed, relatively small, is often volatile. It is sometimes referred to as "Main Memory". It can be accessed immediately and randomly. Primary memory is divided into RAM and ROM.

Secondary memory, also known as peripheral storage, is where the computer stores information that is not currently in use. It is typically slower and has higher capacity than primary storage. It is non-volatile.

2.4.1. Primary Memory:

Random Access Memory (RAM):

The primary storage is referred to as random access memory (RAM) because it is possible to randomly select and use any location of the memory, directly store and retrieve data. It is also called read/write memory. The storage of data and instructions inside the primary storage is temporary. The contents of RAM are lost as soon as the power to the computer is switched off or if there is a minor fluctuation in the power supply. The memories, which lose their content on failure of power supply, are known as **volatile** memories. There are different types of RAM:

Static RAM (SRAM): This RAM will maintain its data as long as power is provided to the memory chips. It does not need to be rewritten periodically. SRAM is often used as cache memory due to its speed.

Dynamic RAM (DRAM): DRAM unlike SRAM, must be continuously re-written in order for it to maintain its data. DRAM is used for most system memory because it is cheap and small.

Read Only Memory (ROM):

Storage of program and data in the ROM is permanent. The ROM stores some standard processing programs supplied by the manufacturers to operate the personal computer. The contents of ROM can only be read by the CPU, but it cannot alter the contents. The basic input/output program is stored in the ROM. It examines and initializes various equipment attached to the PC. The memories, which do not lose their content on failure of power supply, are known as **non-volatile** memories. ROM is non-volatile memory.

PROM (PROGRAMMABLE READ ONLY MEMORY): We know that it is not possible to modify or erase programs stored in ROM, but in PROM it is possible for us to store our program in PROM chip. Writing of programs in PROM can only be done once.

EPROM (ERASABLE PROGRAMMABLE READ ONLY MEMORY): This memory chip overcomes the problem of PROM and ROM. EPROM chip can be programmed time to time by erasing the information stored earlier in it. Information stored in EPROM is erased by exposing it to ultraviolet light. It can be reprogrammed using a special programming facility.

EEPROM: Stands for Electrically Erasable Programmable Read Only Memory. This ROM can be written through the use of a special software program.

2.4.2. Secondary Memory:

Secondary memories are also known as Auxiliary memory. It is the memory supplement to main storage. This is a non-volatile memory. The term non-volatile means it stores and retains the programs and data even after the computer is switched off. The operating speed is slower than that

of the primary storage. Huge volumes of data are stored here on permanent basis and transferred to the primary storage as and when required.

The most common types of auxiliary storage devices are magnetic tape, magnetic disk, floppy disk, hard disk, and optical disk. These devices are divided into two categories based on the type of the access: **Sequential and Random**. Magnetic tapes are examples of sequential accessing devices in which the data can be accessed one after another in a sequential order. In contrast, disks are random-access devices in which the data accessing is direct. Other examples of direct access devices are magnetic disks, optical disks, zip disks etc.

Magnetic Tape:

Magnetic tapes are used for large computers like mainframe computers where large volumes of data is stored for a longer time. In PC we can also use tapes in the form of cassettes. The cost of storing data in tapes is inexpensive. Tapes consist of magnetized material that store data permanently. It similar to cassette tape used in a tape recorder. Accessing data on tapes is much slower than accessing data on disks. Tapes come in variety of sizes and formats.

Magnetic Disk:

Hard disk is a magnetic disk on which we can store computer data. Hard disks hold more data and are faster than floppy disks. A single hard disk usually consists of several platters. Each platter requires two read/write heads, one for each side. It rotates at very high speed. Data is stored on both the surfaces 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.

Floppy Disk:

It is similar to magnetic disk. Floppy disks are portable, because we can remove them from a disk drive. Disk drives for floppy disks are floppy drives. Floppies come in two sizes: They are 5.25 inch or 3.5 inch in diameter. The capacity of a 5.25-inch floppy is 1.2 mega bytes(1.2MB) whereas for 3.5 inch floppy it is 1.44 mega bytes(1.44MB). It is cheaper than any other storage devices. Floppy is a low cost device particularly suitable for personal computer system.

Optical Disk:

Optical disks are a storage medium from which data is read and to which it is written by lasers. Optical disks can store much more data – upto 6GB. There are three basic types of optical disks. They are:

- Compact Disk/ Read Only Memory (CD-ROM)
- Write Once, Read Many (WORM)
- Erasable Optical Disk

2.5. Cache Memory:

The speed of CPU is extremely high compared to the access time of main memory. Therefore the performance of CPU decreases due to the slow speed of main memory. To compensate the mismatch in operating speed, a small memory chip is attached between CPU and main memory whose access time is very close to the processing speed of CPU. It is called **cache** memory. **Cache** memories are accessed much faster than conventional RAM. It is used to store programs or data currently being executed or temporary data frequently used by the CPU. So each memory makes main memory to be faster and larger than it really is. It is very expensive to have bigger size of cache memory, its size is normally kept small.

2.6.Registers:

The CPU processes data and instructions with high speed. There is also movement of data between various units of computer. It is necessary to transfer the processed data with high speed. So the computer uses a number of special memory units called *registers*. They are not part of the main memory but they store data or information temporarily and pass it on as directed by the control unit.

2.7 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. Following are the most useful input devices.

Keyboard:

The most often used input device is the **keyboard**. This is the standard input device attached to all computers. The layout of keyboard is just like the traditional typewriter. It also contains some extra command keys and function keys. It contains a total of 101 to 104 keys.

Mouse:

Mouse is an input device that can be connected to our personal computer. It rolls on a small ball and has two or three buttons on the top. When you roll the mouse on a flat surface, you control the position of the computer's cursor, or pointer, on the screen. As you slide the mouse forward, for instance, the cursor moves forward to the top of the screen. With many programs, the computer conveys your commands to the computer via on-screen pictures or icons. The cursor moves very fast with mouse giving you more freedom to work in any direction.

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 recognize the characters on the cheque.

Optical Mark Reader (OMR): This technique is used for evaluation of objective type answers. The answering will be done by darkening a square or circular space by pencil. The OMR is used to read the answers on the sheet and computer compares them with the answers in the memory for evaluation.

Optical Character Recognition (OCR):

This technique unites the direct reading and comparing of any printed character. Suppose we have a set of hand written characters on a piece of paper, the OCR is used to read and compare these with a set of patterns stored inside the computer.

2.8 Output Devices:

Visual Display Unit (VDU):

The most popular input/output device is the Visual Display Unit (VDU). It is also called the monitor. Monitor is used to display the input data and to receive messages from the computer. The contents of the memory can be viewed on a monitor. A monitor has its own box, which is separated from the main computer system and is connected to the computer by a cable. In some systems it is compact with the system unit. It can be *color* or *monochrome*.

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 can be classified as dumb terminals or intelligent terminals depending upon the work situation.

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 graphics.

2.9. Summary:

In general, a computer is a machine, which accepts data processes it and returns processed information as output. As you become familiar with software and hardware we will have a better understanding of technology for business and for our home. Technology is a vast field and learning it will be a life long process.

In this lesson we discussed five basic operations that a computer performs. That is input, storage, processing, output and control. A computer accepts data as input, stores it, processes it as the user requires and provides the output in a desired format. The storage unit of a computer is divided into two parts: primary storage and secondary storage. All computers have several parts in common:

- Input devices allow data and instructions or commands to be entered into the system.
- A means of storing instructions and data.
- A central processing unit (CPU) which controls the processing.
- A means of returning the processed information in the form of output.

2.10. Technical Terms:

Bit: stands for **binary digit**. The smallest piece of computer information, either the digit 0 or 1.

Binary code: The most basic language a computer understands, it is composed of a series of 0s and 1s.

Memory: storage for information, including applications and documents.

Program: A series of instructions written in a computer language that tells the computer what to do and how to do it.

I/O: Transfer of data into a computer, and from the computer to the outside world.

Flash memory: A small printed circuit board that holds large amounts of data in memory.

Laser :(light amplification through stimulated emissions of radiation)

A device that uses the oscillation of atoms or molecules between energy levels to generate a narrow, finely focused beam of light. Lasers are used in fiber optics, laser printers, CD-ROM players, and scanners.

2.11. Model Questions:

1. What are the basic operations performed by any computer system?
2. Draw a block diagram of a computer? Explain the function of various units.
3. Differentiate between RAM and ROM. Also distinguish between PROM and EPROM?
4. Write about cache memory and registers?
5. Write short notes on
 - (a) ALU
 - (b) Random Access Memory (RAM)
6. Explain the significance of input, output devices in a computer. List various input, output devices that you know?

2.12. References:

Prof.G.S.Prasad, M.Sc, M.S “**Fundamentals of Computer - Telugu Academy**”

Dharma Raja Raman “**Computer Primer**”

Alexis Leon & Mathews Leon “**Fundamentals of Information Technology**”

A.Jaiswal “**Fundamentals of computers and Information technology**”

N.Subramainian “**Computer Genesis, Programming & applications**”

G. RAMANJIAH.,M.Sc (Computers)

Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson 3 :

COMPUTER LANGUAGES AND OPERATING SYSTEM CONCEPTS

3.0 Objectives:

The main objectives of this lesson are:

- To know the concept of software.
- To understand different types of software.
- To know what a language is.
- To know about different types of translators.
- To know what an Operating System is.
- To know about Booting process
- To understand various types of operating systems.
- To know the basic concepts in DOS
- To understand Internal and External commands.

Structure of the Lesson:

- 3.1. Introduction**
- 3.2. What is software?**
- 3.3. Computer languages**
- 3.4. Translators**
- 3.5. What is an operating system**
- 3.6. Types of operating systems**
- 3.7. Introduction to DOS**
- 3.8. DOS internal commands**
- 3.9. DOS external commands**
- 3.10. Summary**
- 3.11. Technical terms**
- 3.12. Model questions**
- 3.13. References**

3.1. Introduction:

In the previous lesson we discussed about the parts and functions of computer. It has been mentioned that programs or commands have to be fed to the computer to do a specific task. So it is necessary to provide sequence of instructions for our work to be done. We can divide the computer components into two major areas, namely, *hardware* and *software*. Hardware is the physical components of a computer system. Software is the set of programs.

3.2. What is Software?

As we know computers will not function without instructions from the user, for a specific job to be accomplished we have to give a sequence of instructions to the computer. This set of instructions or commands is called a computer *program*. Software refers to a set or group of computer programs, procedures that describe the programs and how they are to be used. The process of software development is called *programming*.

Both software and hardware have to work together to produce meaningful result. To develop a software is difficult and expensive.

Computer software is normally classified into two broad categories.

- System software
- Application software

System Software: System software may be defined as a set of one or more programs designed to control the operation of the computer system. System software also facilitates the programming, testing, and debugging of computer programs.

Application Software: Application Software is a set of programs to carry out operations for a specific application. For example, railway reservation system software, hotel management software, payroll software, etc., Application software is useful for word processing, billing system, accounting, producing statistical report, analysis of numerous data in research, weather forecasting, etc.

3.3. Computer Languages

Language is a system of communication between two people. Some of the basic natural languages that we are familiar with are English, Hindi, Telugu etc.,. These languages are used to communicate among various categories of people.

Computer will not understand any of the natural languages. So we need a language to communicate with computer. There are programming languages specially developed so that we could pass our data and instructions to the computer to do specific job. Some of these languages

are PASCAL, FORTRAN, BASIC, COBOL, C, C++, VC++ etc. So instructions or programs are written in a particular language based on the type of job.

Programming Languages:

There are two major types of programming languages. These are Low Level Languages and High Level Languages. Low Level languages are further divided into *Machine language* and *Assembly language*.

Low Level Languages

The term low level means closeness to the way in which the machine can understand.

Machine Language

Machine Language is the only language that can be directly understood by the computer. It does not need any translator program. We also call it machine code and it is written as strings of 1's (one) and 0's (zero). When this sequence of codes is fed to the computer, it recognizes the codes and converts it into electrical signals. For example, a program instruction may look like this:

1011000111101

It is not an easy to learn and write instructions in this form. Because of the complexity very few people can write the programs in this language. It is considered to be the first generation language.

The only advantage is that program written in machine language run very fast because no translation is required for the CPU.

Assembly Language:

A low-level programming language that is slightly more user-friendly than machine language. The software which translates a program written in assembly language to machine language is called an assembler. These are considered to be second-generation languages.

Advantages of assembly language are:

It is easier to understand and write programs which are nearer to English language. It saves a lot of time for the programmer, easier to correct errors and modify the program instructions whenever necessary.

High-level Language:

These are third-generation or 3GLs. These are closer to so-called natural languages (we talk). High-level languages are simple languages that use English and mathematical symbols like +, -, %, / etc. for its program construction. Any program written in High-level language has to be converted to machine language for the computer to understand.

Higher-level languages are problem-oriented languages because the instructions are suitable for solving a particular problem. Choice of a particular high-level language depends on the application/problem. Advantages of High Level Languages are easy to learn and use because that they are similar to the languages used by us in our day-to-day life.

3.4. Translators

Translator:

Translator is a program, which converts source language to object language.

Assembler:

Assembler is a translator or converter, which converts assembly language program (Mnemonics or symbols) to machine understandable code called machine language (0s and 1s).

Assembler needs two passes to convert assembly language program to machine language program.



Compiler:

The programs written by the programmer in high-level language is called *source program*. After this program is converted to machine language by the compiler it is called *object program*.

Compiler translates the instruction of a higher-level language to machine language. Thus compiler is a program translator like assembler but more sophisticated. It scans the entire program first and then translates it into machine code.

Every high-level language has its own compiler. For example FORTRAN compiler will not compile source code written in COBOL language.



Interpreter:

An interpreter is another type of program translator used for translating higher-level language into machine language. It takes one statement of high-level language, translates it into machine language and immediately executes it. Translation and execution are carried out for each statement.

It differs from compiler in the following aspects:

Compiler translates the entire source program into machine understandable form at a time and generates an object code which can be used on repeated execution of the program.

With interpreter the program needs to be retranslated every time you want the program to be executed.

3.5.What is an operating system?

Operating system is software, which is an interface between the user of the computer and computer hardware.

The following are the various operating systems: MS-DOS, Windows 95/98/2000/me/NT/XP, UNIX, LINUX etc.

3.6.Types of operating systems:

Basically there are two types of operating systems, single user operating system and multi user operating system.

Single user operating system:

Single user operating system allows one program to execute at a time. The program must completely finish executing before the next can begin.

Multi user operating systems:

A multi user operating system allows a single CPU to execute what appears to be more than one program at the same time, when, in fact only one program is being executed. The CPU switches its attention between programs as it receives requests for processing, executing statements from one program, and then from another. This processing happens so fast that it appears that the number of users exceeds the operating system's capability; noticeable delays in processing will result. Several other operating systems are:

Parallel System:

Some large computers may have more than one processor. In parallel processing systems, tasks are shared between a numbers of processors. The operating system allocates instructions to each processor. The more processors that are allocated to a task - the faster that task will be completed.

Distributed systems:

In a distributed system, computers and peripherals are linked together. Each user is not aware of which computer or peripheral he is using. A user on one computer may be running a program stored on another computer accessing data stored on yet another computer. Here the operating system will control access to the resources.

Real Time systems:

Real-time means that data is processed immediately. The computer used in a real-time system is 'dedicated' - it does nothing else. An example of a real-time system is a Process-Control System. Input data received from sensors is processed immediately, analyzed and any necessary actions taken without any delay.

3.7.introduction to Disk Operating System [DOS]:

When the computer starts, the operating system takes control of the machine. An Operating System is a group of programs that controls and manages the computer Hardware and the Software resources of a computer system.

ABOUT MS-DOS

Microsoft DOS (Disk Operating System) is a command line user interface. MS-DOS 1.0 was released in 1981 for IBM computers and the latest version of MS-DOS is MS-DOS 6.22, which was released in 1994.

DOS operating System is used for operating the system or the computer. The main functions of DOS are to manage disk files, allocate system resources according to the requirement. DOS provides features required to control hardware devices such as Keyboard, Screen, Disk Devices, Printers, and other programs.

Basically, DOS is the interface or a medium through which the user and external devices are communicated. DOS converts the user commands into a computer understandable format. It also translates the result and any error message in user understandable form.

Commands in DOS:

These are the instructions, which are used to communicate with the computer system. We call these instructions are commands. The commands of the DOS are divided into two categories: Internal Command and External Commands

Internal Command:

These are the commands, which reside in a portion of the computer's memory and are loaded along with the operating system into the memory. These commands are always available for execution. An Internal command, which is a command embedded into the command.com file.

External Command:

These are the commands, which have to be loaded from the disk into the memory of the computer before we want to execute them. These commands are usually in the form of file, which resides in the disk, as and when required they will be loaded into memory before execution.

3.8. DOS Internal Commands:

CLS:

CLS command allows a user to clear the complete contents of the screen and leave only a prompt.

Syntax: CLS

VOL:

VOL command displays the volume information about the designated drive.

Syntax: VOL

VER:

VER command displays the version of MS-DOS or if running Windows 95 or above the version of Windows.

Syntax: VER

TIME :

Allows the user to view and edit the computer system time.

Syntax: TIME [time] Displays or sets the system time.

Type TIME with no parameters. Displays the current time setting and a prompt for a new one. Press ENTER to keep the same time.

DATE:

The date command can be used to view the current date of the computer as well as change the date.

Syntax: Displays or sets the date.
DATE [date]

Type DATE without parameters. Displays the current date setting and a prompt for a new one. Press ENTER to keep the same date

DEL:

DEL is a command used to delete files from the memory of the computer.

Syntax: Deletes one or more files.
DEL [drive:][path]filename [/P]

Ex: del ramu.txt

REN:

Used to rename files and directories from the original name to a new name.

Syntax: REN old name new name
Ex: REN RAMU.TXT HANUMAN.TXT

CD :

CD (Change Directory) is a command used to switch directories in MS-DOS. The Change Directory (CHDIR) or CD command enables the user to travel around the directories in a disk. Type the CD command at the command prompt.

Syntax: CD[drive:][path]
CD[.] cd.. - Goes back one directory.

Ex: CD MAIN
CD..

DIR:

The dir command allows us to see the available files in the current and or parent directories.

Syntax: Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[:]attributes]]

/p - pauses after each screen full of information.

/w - uses wide list information.

/o - list of files in sorted order.

/l - uses lowercase letter.

Ex: dir/p

COPY:

Allows the user to copy one or more files to an alternate location.

Syntax:: Copies one or more files to another location.
COPY [/A | /B] source [/A | /B] [+ source [/A | /B] [+ ...]] [destination] [/A | /B]]
[/V] [/Y | /-Y]

Ex:

User can create a file using the copy command. In the below example we create the file called "test.txt".

copy con test.txt

Once the above command has been typed in a user could type in whatever contents he or she wishes. When you have completed creating the file you can save and exit the file by pressing and holding **CTRL key and pressing Z key** and then press enter. A simple alternative would be pressing F6 functional key.

RD:

Removes empty directories in MS-DOS.

Syntax: Removes (deletes) a directory.
RMDIR [drive:]path
RD [drive:]path

Ex: RD MAIN

3.9.DOS External Commands:

CHKDSK:

chkdsk is a utility, which checks the computers hard disk drives status for any cross-linked or any additional errors with the hard disk drive. It checks for the memory and gives the detailed information about the memory consumed as well as free memory.

DISKCOPY:

The disk copy utility allows a user to copy the complete contents of a diskette to another diskette

DOSKEY:

Doskey is a MS-DOS utility that allows a user to keep a history of commands used on the computer. This allows frequently used commands to be used without having to type the commands in each time you need to use it.

EDIT:

Edit allows a user to view, create and or modify their computer files.

FDISK:

Fdisk allows the user to delete and/or create partitions on the hard disk drive.

FORMAT:

Format is used to erase all of the information off of a computer diskette or fixed drive.

LABEL:

Label is used to view or change the label of the computer disk drives.

TREE:

This command is used to display all the directories and also the files if specified.

3.10.Summary:

In this lesson we discussed about two types of software, namely, system software and application software. It is designed for performing tasks such as controlling all operations required to move data into and out of the computer. It communicates with printer, card reader, disk, tapes, etc. and monitors the use of various components like memory, CPU, etc. Application software is a set of programs written for specific purpose. We have discussed about levels of computer language.

In this lesson we discussed very important software component called operating system. Operating system is system software, which provides a link between the user of a computer and hardware. It controls all the activities of the computer system. There are different types of operating systems: single user, multi user, parallel, distributed and real time operating systems. In this lesson we discussed introduction to DOS, internal and external commands.

3.11.Technical terms:

CPU: Central processing unit of a computer. Its consists of three sub-units: ALU - Arithmetic and Logical Unit

CU - Control Unit

MU – Memory Unit (Main memory)

FORTRAN: FORmula TRANslation, a high level language used for scientific and engineering applications.

MS-DOS: an operating system designed by Microsoft corporation for IBM compatible personal computers.

Object Program: the machine language program resulting from the translation of a user(source) program.

Source program: A program written following the syntax of a language.

Time-shared system: A computer system with an operating system designed to allow many users to simultaneously use it.

3.12. Model Questions:

1. What is software and hardware? Differentiate application and system software?
2. What is computer Language? Explain the three different categories of computer languages.
3. What is a translator? Explain different types of translators or converters?
4. What is interpreter? How does it differ from compiler?
5. What is an operating system (OS)? Explain different types OS?
6. What is booting a computer? Explain types of booting?
6. Write about DOS internal commands?
7. Write about DOS external commands?

3.13.References:

Prof.G.S.Prasad, M.Sc, M.S “**Fundamentals of Computer - Telugu Academy**”

Dharma Raja Raman “**Computer Primer**”

Alexis Leon & Mathews Leon “**Fundamentals of Information Technology**”

A.Jaiswal “**Fundamentals of computers and Information technology**”

N.Subramainian “**Computer Genesis, Programming & applications**”

G.RAMANJIAH.,M.Sc(Computers)
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson 4

WINDOWS OPERATING SYSTEM CONCEPTS

4.0 Objectives:

At the end of the lesson the student must be able to understand

- Windows OS
- Features of windows.
- Working with windows, folders etc.

Structure of the Lesson:

- 4.1. Introduction
- 4.2. Versions of windows operating system
- 4.3. Features of windows operating system
- 4.4. Overview of windows Desktop
- 4.5. Icons on the Desktop
- 4.6. Parts of a window
- 4.7. Mouse operations
- 4.8. File commands
- 4.9. File attributes
- 4.10. Special indicators in the menu
- 4.11. Start button
- 4.12. Overview of the start button
- 4.13. Taskbar properties
- 4.14. Clock settings
- 4.15. Shutdown menu
- 4.16. Find menu
- 4.17. Settings menu
- 4.18. Document folder
- 4.19. Programs menu
- 4.20. Windows explorer
- 4.21. Screen properties
- 4.22. Summary
- 4.23. Technical terms
- 4.24. Model questions
- 4.25. References.

4.1.Introduction

The term "windows" also refers to a program that takes advantage of the computer's graphics capabilities to make the computer easier to use. This is often called Graphical User Interface or GUI.

Windows 98 is an operating system (OS). An operating system is the software that runs on your computer to control the basic operation of your computer's hardware components such as the processor, monitor, keyboard, mouse, memory, drives, printers, scanners, etc. Applications such as Microsoft Word and Excel must run "on top" of the operating system.

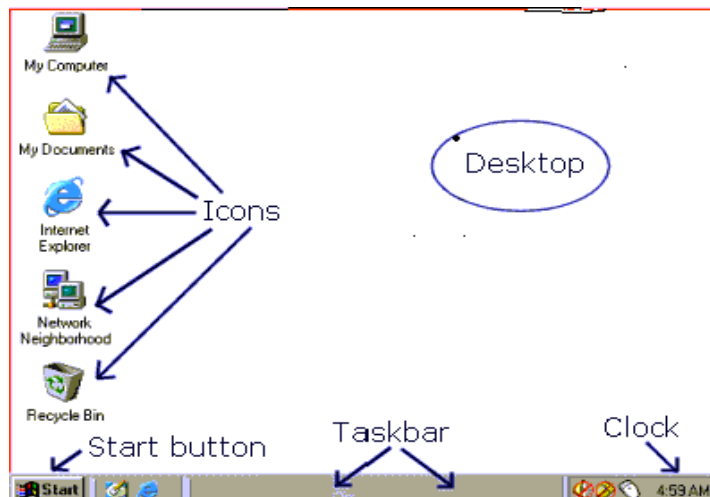
Microsoft windows is the popular GUI based operating system from Microsoft Corporation, offers an excellent platform and environment for the user. This is very user-friendly environment. Windows has several versions windows 3.x/ 95/98/me/2000/NT/XP. Windows 95 and 98 are single user operating systems and where as Windows NT / servers are multi user operating system.

4.2.Features of Windows:

- Windows provides user-friendly environment.
- It is an easy-to-use, consistent Graphical User Interface environment.
- It is capable of multitasking of graphical programs and character based programs.
- Program manager can be used to switch easily between multiple programs.
- Built in accessories such as wordpad and paint.
- A powerful help facility is provided which will enable a user to sort out its problems in comfortable manner.
- For entertainment, window comes with some games.
- Searching for specific files is very easy.
- It provides multimedia related applications.

4.3.Overview of Windows Desktop:

When the computer is switched on, windows starts automatically. It displays a login screen where the user name and password for using the network services has to be entered. On entering the correct password, the user will be able to use network services. If the user clicks on Cancel button, user will not able to use the network services. Then it shows the Desktop.



The Windows Desktop

Desktop:

The entire screen is called the desktop. It is the first view after entering the password in the login screen. The pictures on the desktop are known as icons. These icons can be arranged in any order in the desktop.

Taskbar:

The rectangular bar that runs horizontally across the bottom of the screen is called taskbar. The position of the taskbar can shift according to the user's requirement. It provides access to the all programs. It includes start button, current time etc.

4.4. Icons on the Desktop

Icons are small graphical images that can represent computer programs, files, folders and printers among other things. To activate an icon you double click on it with the left mouse button, this will activate the icon and either start a program or open a file/folder.

The icons on your desktop can be renamed by right clicking on them and selecting **rename**, similarly they can be deleted by right clicking and selecting **delete**.

My Computer:

This is a standard icon and is very useful; it contains all the resources available in the computer. Generally it consists of all the drives in the computer, control panel, printer folder. To change the view of the contents in my computer, we use the various options in the view menu.

My Documents:

The My Documents icon links to a standard folder, which Windows 98 uses to store the documents. By default Windows 98 will save any documents we create in this folder.

Internet Explorer:

This icon is a shortcut to Microsoft's Internet Explorer program. Internet Explorer is used to view web sites and web pages on the Internet.

Network Neighborhood:

The Network Neighborhood Icon represents a folder that contains links to any other computers that are in workgroup. The contents of network neighborhood are workgroups, servers, share folder and printers.

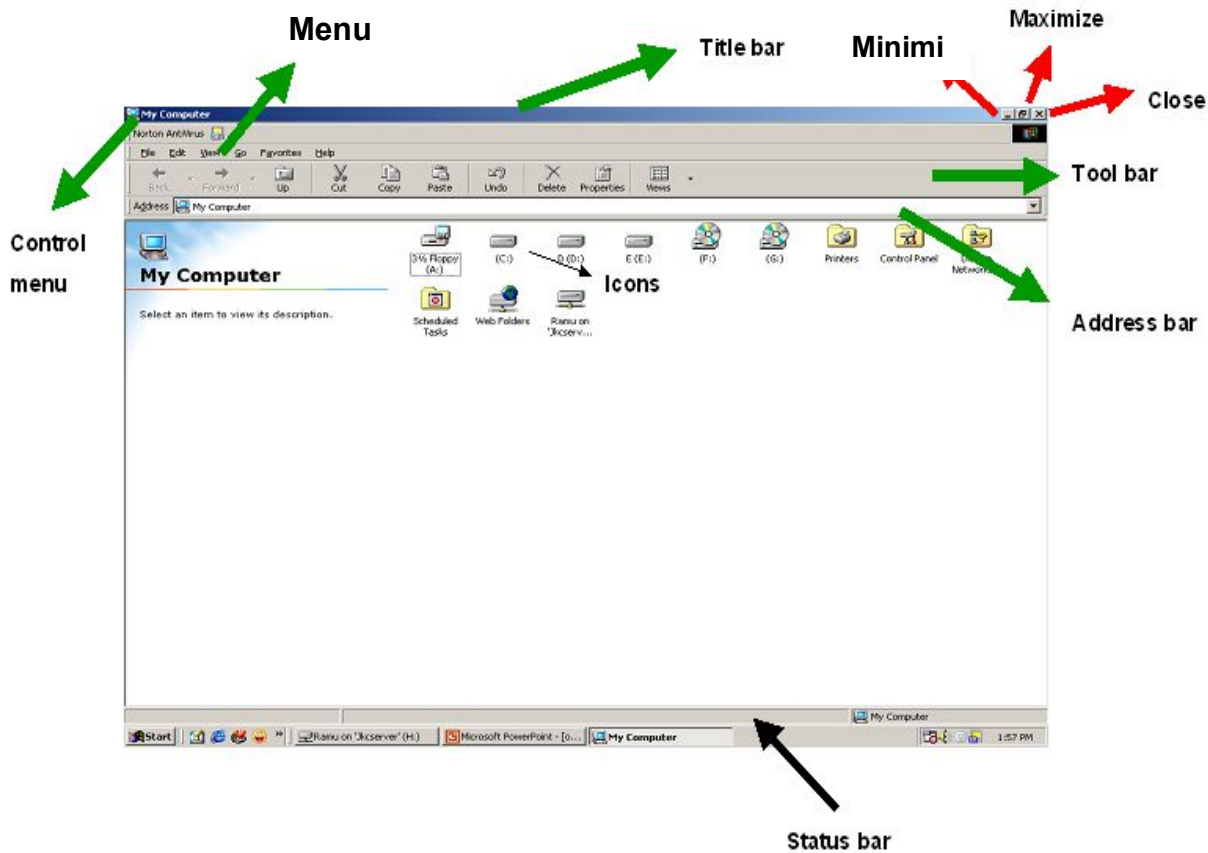
Recycle Bin:

All files and folders, which are deleted from the windows, are stored in the recycle bin. By default windows 98 is setup to deposit all deleted files, folders or icons in the recycle bin. Recycle bin can be used to restore files that have been deleted.

4.5.Parts of a Window:

Every computer program window looks a little different because each has its own menus, icons, and controls.

Windows has Title bar, Control menu, Minimize box, Maximize/restore box, Close button, Status bar, Menu bar.



Title bar:

The *title bar* is the horizontal bar at the top of a window that shows the name of the program and document. You can move window around your desktop. The title bar is highlighted when the window is active.

Control menu:

This has list of commands, usually contains minimize, maximize, move, size etc.

Minimize button:

Click the *Minimize button* to reduce the active window to a button on the taskbar using minimum space on the screen.

Maximize Button:

Click the *Maximize button* to expand the active window to fit the full screen. Click the maximize button again to restore the window to the smaller size.

Resizing Button:

Only the window, which is maximized, can be resized.

Close Button: Click the *Close button* to close the active window or program.

Menu Button: Consists of a set of keywords (control words) for this window.

Status Bar: Shows the comment on selected item.

Menu bar: Window menu is a list of commands that you can select. Click a menu bar item to perform a command in a program.

4.6. Mouse operations:

The mouse is used to move around and activate different features of Windows applications in the following manner

Click:

Press and release the left button on the mouse.

Right Click:

Point the mouse and then clicking the right button to display the short-cut menu for that item.

Double click:

Point the mouse to an item of choice and click the left button twice, rapidly. This is used to start programs and open files.

Drag & Drop:

Hold down the left mouse button of the mouse while you are moving the mouse and release it at the desired position.



4.7. File Commands:

Copying:

Copying files to another directory or drive. Select explorer: Select the files to be copied, hold the ctrl key, and Drag the selection to the destination.

Deleting:

Deletes selected files. Select the files and press Del or File Delete, Delete dialog box appears showing the current directory and the name of the file to be deleted. Click on Yes button. The deleted files are sent to recycle bin.

Renaming:

Renames selected files. Select the item to be renamed, Choose File -> Rename, Current name get selected, and Type the new name. Press enter or click outside the file name.

Moving:

Move the files to another directory. Select the files to be moved and then drag them as a group to the new destination.

4.8.File Attributes:

Select the files whose attributes you want to alter. Choose File -> Properties then Properties Dialog box appears, check the boxes appropriately. The attributes are:

Read only: Does not allow any one to make any changes to the file.

Archive: Indicates the file has been modified since the last backup. This is used by backup program such as the DOS backup command.

Hidden: Hides the file from the directory listing in DOS and directory Windows in the explorer.

System: To tell windows that it is an Operating system file. Move the mouse pointer to the menu item in the menu bar and then click on it.

Closing menu: Click on the Menu or anywhere outside the menu. Choosing a menu item, Click on the items using mouse.

4.9.Special Indicators in Menu:

Window menu often have special symbols that tell you more about Menu Commands. They are

Dimmed command name: If a menu command is dull or gray in color it means that this choice is not currently available.

Ellipses(...) : If the window or window application requires more information from your end before the execution of a particular command, the names of such commands shall be followed by Ellipses.

Triangle(▶): A triangle preceding the command has additional sub-choices for you to be made. Selecting such command brings up a cascading menu with more options to chosen from.

Option buttons(•): There are exclusive settings. They are round boxes, and only one can be set on at a time. To select an option button click on the option button.

4.10.The Start Button:

The Start button is the single most commonly used feature in windows. . This button is the primary thing of starting programs from the desktop. This is a very important part of Windows. Using the Start button you can start programs, open documents, customize your system, get help, search for items on your computer, and more.

To open a start menu item, first click Start. The start menu opens showing you the first level of menu items. When you move the mouse pointer over any menu item that has an arrow next to it, another menu, or submenu, opens and reveals related files, programs, or commands. Click on a file or program to open it. The Start Button gives you access to programs that you have loaded on the computer. It also lets you access common tools that come with Windows.

4.11.Overview of the Start menu:

Programs:

This section of the Start Menu holds all of the program groups and programs installed on your machine. You can open the programs menu by pointing to it with the mouse.

Favorites:

This Folder keeps track of your favorite files: games, programs, web-sites, you name it. Click on any item in the list to automatically go there.

Documents:

The Documents section contains the last few documents you worked on. By clicking on a document name, the document will be opened in the program that was used to create it.

Settings:

The Settings section enables you to change the settings and basic appearance of Windows 98.

Find:

Use the Find feature to search a drive, multiple drives, or your entire system (including networked drives) for a folder or file you are searching for.

Help

Help starts the Windows 98 help program. This program can also be accessed by pressing F1. See the Help section for more information

Run

Run can be used to start a program on your hard drive or on a floppy disk. To start a program, either type in the name of the program, or choose Browse to select the file to run

Log Off

Use this feature if you want to log off your system and allow someone else to log on without running through a complete shut down of the system.

Shutdown

The Shutdown menu will allow you to turn the computer off or reboot your system as you choose.

4.12.Taskbar Properties:

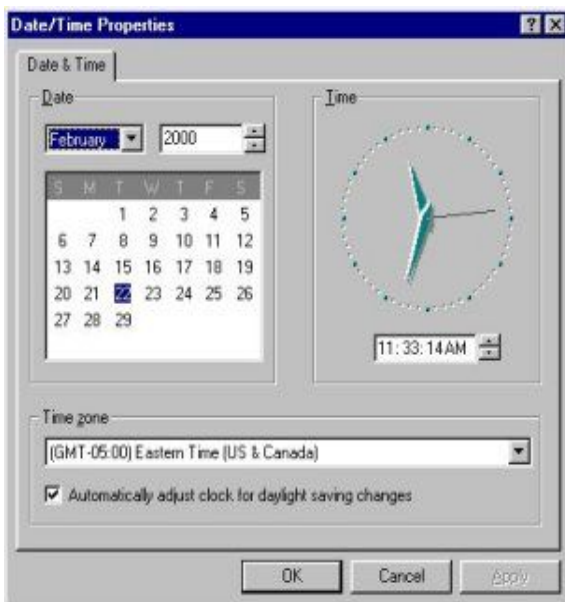
Click the start button. Put to setting on Start menu.
On the setting menu click on the Taskbar.
Select the Taskbar options tab.
You can change the following settings, using the options tab.

- Always on top
- Auto Hide
- Show small icons in Start menu
- Show clock



4.13.The Clock - Setting The Date and Time:

If the date or time is incorrect, right click on the clock and select adjust date/time.



Select the correct year, month and day from the calendar area on the left side of the window. On the text box, which displays the digital clock, using the left mouse button, single click to place the cursor in the area which is incorrect, hh, mm, ss, am/pm.

Use the scroll areas to the right to increase or decrease the time, which ever is applicable. When the date and time is correct, press the OK button to exit. Other clock properties can be found in the Regional Settings option found in the Control Panel.

4.14. The Shutdown menu:

We should never just turn off the computer. To avoid damage to the system or loss of files, always follow specific steps to **shut down** the computer. The **Start** menu is where you begin the shutdown process. It has several options they are:

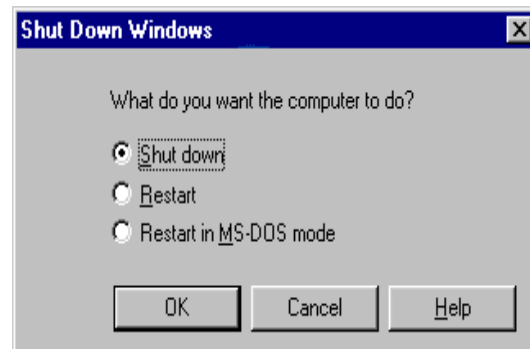
Shuts down the PC, always use this option to your PC.

Restart:

This will restart the computer.

in MS-DOS Mode:

This will cause the PC to restart in the MS-environment.



turn off

Restart

DOS

4.15. The Find menu:



Files or Folders

This is used to search for files or folders on your PC (see related pages at top of this page for a guide).

Computer

This is used to find computers on your network.

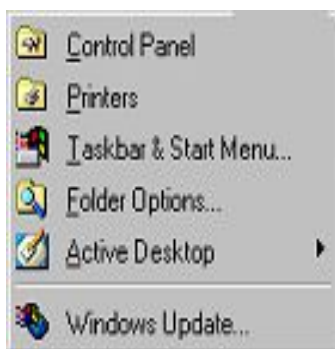
On the Internet

This will search the Internet for your query.

People

This will search your PC and other sources for people.

4.16. The Settings Menu:



Control Panel - The control panel is used to configure Windows 98.

Printers - Used to add and configure printers.

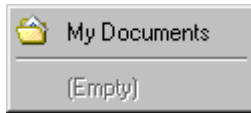
Taskbar & Start Menu - Here you can customise your taskbar & start menu.

Folder Options - Here you can change the look and feel of your folders.

Active Desktop - Changes the desktop settings.

Windows Update - starts the automated windows update.

4.17. The Documents folder:



This acts like a folder, it will contain the recently viewed documents for fast access as well as a link to the main **My Documents** folder.

4.18. The Programs Menu:

To open a program simply left click on it, as with the start menu the options with the small black arrow to the right hand side contains another sub menu. For example if you wanted to open a program in the **accessories** folder, then you would hover the mouse over accessories and then click on the program you want from the sub menu that appears. You can rename any program on the menu by right clicking on it and selecting **rename** from the menu that appears. Whenever you install software it will (normally) appear on the programs menu, if you ever want to remove a program then use the control panel to uninstall it, simply deleting it from the programs menu will not uninstall the software.

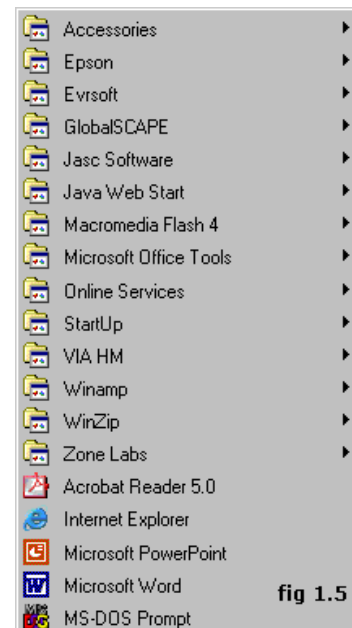
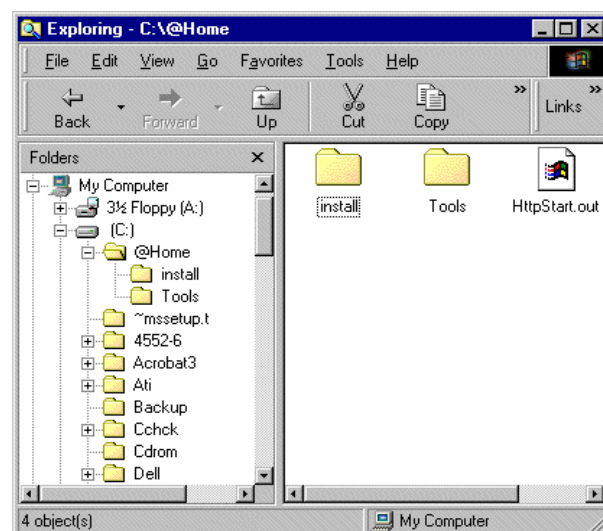


fig 1.5

4.19. What is Windows Explorer?

Windows Explorer is a file management program similar to File Manager in Windows 3.1. It allows customers to move or copy files from one folder or drive to another, and to safely delete files by sending them to the recycle bin. In addition, Windows Explorer allows you to view Web pages directly in the window.

Windows98 separates the window into two panes. If you click on an object in the left pane, the contents of the object displays in the right pane. Click on Desktop and the contents of the Desktop folder display on the



right. Click on My Computer and your computer resources are displayed on the right. To see the contents of a drive, click on the drive. To see the contents of a folder, click on the icon for the folder in the left pane.

4.20. Screen Properties:

To change or access options for the "look" of the desktop, right click on the desktop area. This area should be free of icons. From the quick menu, which appears, select "Properties".

Background:

The background options allows your desktop to have fancy patterns and wallpaper for your viewing pleasure.

Screen Savers:

It provides some thing to do while the computer is idle. If the computer is idle for a set period of time, the screen saver will automatically begin. To stop the screen saver, simply move the mouse, or press any key on the keyboard. It will offer some security features (Password) also.

Appearance:

The appearance option is used to set the system colors for items such as desktop background, title bars, menus etc. Use caution when selecting colors to ensure visibility of text.

Settings:

The settings option is used to change the viewing area of the desktop, setting font size, and size of the desktop area. These settings can control one monitor in Windows '95, and two monitors in Windows '98. They should only be changed by experienced users.

4.21. Summary:

In this lesson we discussed different type of operating system including windows 3.x/95/98/NT/XP. Windows is a popular Graphical user interface (GUI) based operating system from Microsoft Corporation, offers an excellent platform and environment for the user. It provides user-friendly environment.

Windows desktop contains several components, icons, start button. Every action should be starting from the start button. We have discussed the start button items, parts of a window, and screen properties like changing desktop wallpaper, screen saver windows explorer and its components.

4.22. Technical Terms:

GUI: Systems software that allows users to have direct control of the visible objects such as icons and actions that replace command syntax.

Multitasking: The operation of two or more programs running on a single processor (CPU) at the same time.

Icon: An icon is a small object used to remind the user of a particular operation, idea or a project.

Window Operating system: An operating system from Microsoft, in a variety of products, that provides a GUI using icons to provide instant access to common task.

Taskbar: A bar on the graphic user interface that shows all open applications and active windows. Clicking any task button on the taskbar brings the corresponding session to the foreground.

Shutdown: The term "shut down" can be used to mean, "turning off" something, but most commonly used for computers.

4.23.Model Questions:

1. What is windows operating system? Explain its features?
2. Explain the parts of a window?
3. Write about windows explorer?
4. Give overview of start button?
5. Write about file commands and attributes in windows?

4.24.References:

Alexis Leon & Mathews Leon “**Fundamentals of Information Technology**”.

A.Jaiswal “**Fundamentals of computers and Information technology**”.

Ron Mansfield “**Working in Microsoft Office**”. Tata McGraw Hill.

Sharon Crawford and Neil J.Salkind “**ABC of Windows 98**” , BPB publications.

G.RAMANJIAH.,M.Sc(Computers)
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson 5

MICROSOFT OFFICE AND WORD PROCESSING

5.0 Objectives:

This lesson provides an introduction to the Microsoft Office. It covers Microsoft Office Products, introduction to Word-Processing and features of Word-Processing.

Structure of the Lesson:

- 5.1 Introduction to Microsoft Office
- 5.2 Word-processing
- 5.3. Features of word-processing
- 5.4. Summary
- 5.5. Technical Terms
- 5.6. Model Questions
- 5.7. References

5.1. Introduction to MS-OFFICE:

Microsoft Office for windows is a suite of applications that are also available as independent programs. Microsoft Office is a major facility in computer applications field. Microsoft Office comes in two ways. One is called **Standard** and other one is called **Professional**. Both the versions have the following primary applications.

1. Microsoft Word
2. Microsoft Excel
3. Microsoft PowerPoint
4. Microsoft Access
5. Microsoft Mail
6. Microsoft Graph
7. Microsoft Clip Art Gallery
8. Microsoft Word Art

9. Microsoft Organization Chart

10. Microsoft Equation Editor

11. Microsoft Publisher

Microsoft Word

This is a full-featured word processor that can create letters, menus, reports, newsletters, manuals and any kind of documents. You can use it for daily correspondence and desktop publishing also. It is having more menus, keyboard shortcuts and toolbars. Toolbars containing buttons that you can click to perform, common tasks like centering text, saving and printing documents. After developing our own working style we can make custom toolbars. Word provides macro capability to do repetitive tasks. It provides some model reports called templates, rather than typing all the matters it gives skeleton reports. Wizards are more powerful than templates. Word having spell checker that checks the document and more features like Auto formatting, Printing envelops and labels, Mail Merge.

Microsoft Excel

This is a spreadsheet program that allows you to organize, analyze information. It is designed to work with numbers. You can enter data in cells arranged horizontal rows and vertical columns on work sheets. Excel organizes the worksheets with workbook. Every workbook having 3 (default) worksheets and you can add up to 255 worksheets and you can remove 1 or 2 worksheets from the three. Worksheets contains 65,536 rows and 256 columns (numbered A to IV)..The address of the cell is combination of row and column labels. Excel provides tool bars, menu choices and functions. Other than these features excel provides Screen wizard, autocalc, auto complete, templates, templates wizard etc

Microsoft PowerPoint

It is a presentation tool. We can create slides, handouts and speaker notes. The slides can be arranged sequentially in presentations.

Microsoft Access

This is a powerful database program but you can store, retrieve, and organize data. Access comes with wizards and variety of predefined database elements.

Microsoft Mail

Microsoft Mail is a product that comes with a license to use mail on the workstation where office is installed. If your workgroup is using Microsoft Mail, this license allows to copy the necessary software to your machine.

Microsoft Graph

Microsoft Graph is used to convert the tabular as well as text data into the graphs and charts. A number of graphs and chart options are provided by this component of Office.

Microsoft Clip Art Gallery

Clip Art Gallery is a collection of ready-made graphics that the user can import into any of the primary applications to make them more effective.

Microsoft Word Art

Word Art enables a user to convert the text into various shapes for logos, banners, and headlines.

Microsoft Organization Chart

Microsoft Organization Chart provides the tools for creating figures such as organization charts by selecting the different styles and organizations for charts.

Microsoft Equation Editor

Microsoft Equation Editor enables a user to format complex equations correctly.

Microsoft Publisher

It helps to organize the material of textbooks in the form of chapters and volumes.

5.2. Word Processing:

Using a computer to create, edit, and print documents is called word processing. Of all computer applications, word processing is the most common. To perform word processing, you need a computer, a special program called a word processor and a printer. A word processor enables you to create a document, store it electronically on a disk, display it on a screen, modify it by entering commands and characters from the keyboard and print it on a printer.

The great advantage of word processing over using a typewriter is that you can make changes without retyping the entire document. If you make a typing mistake, you simply back up the cursor and correct your mistake. If you want to delete a paragraph, you simply remove it, without leaving a trace. It is equally easy to insert a word, sentence, or paragraph in the middle of a document. Word processors also make it easy to move sections of text from one place to another within a document, or between documents. When you have made all the changes you want, you can send the file to a printer to get a hardcopy.

5.3. Features of word processing:

Insert text: Allows you to insert text anywhere in the document.

Delete text: Allows you to erase characters, words, lines, or pages as easily as you can erase them on a black board.

Cut and paste: Allows you to remove (cut) a section of text from one place in a document and insert (paste) it somewhere else.

Copy: Allows you to duplicate a section of text.

Page size and margins: Allows you to define various page sizes and margins, and the word processor will automatically readjust the text so that it fits.

Search and replace: Allows you to direct the word processor to search for a particular word or phrase. You can also direct the word processor to replace one group of characters with another everywhere that the first group appears.

Word wrap: The word processor automatically moves to the next line when you have filled one line with text, and it will readjust text if you change the margins.

Print: Allows you to send a document to a printer to get hardcopy.

File management: Many word processors contain file management capabilities that allow you to create, delete, move, and search for files.

Font specifications: Allows you to change fonts within a document. For example, you can specify bold, italics, and underlining. Most word processors also let you change the font size and even the typeface.

Footnotes and cross-references: Automates the numbering and placement of footnotes and enables you to easily cross-reference other sections of the document.

Graphics: Allows you to embed illustrations and graphs into a document. Some word processors let you create the illustrations within the word processor; others let you insert an illustration produced by a different program.

Headers, footers, and page numbering: Allows you to specify customized headers and footers that the word processor will put at the top and bottom of every page. The word processor automatically keeps track of page numbers so that the correct number appears on each page.

Layout: Allows you to specify different margins within a single document and to specify various methods for indenting paragraphs.

Macros: A macro is a character or word that represents a series of keystrokes. The keystrokes can represent text or commands. The ability to define macros allows you to save yourself a lot of time by replacing common combinations of keystrokes.

Merges: Allows you to merge text from one file into another file. This is particularly useful for generating many files that have the same format but different data. Generating mailing labels is the classic example of using merges.

Spell checker: A utility that allows you to check the spelling of words. It will highlight words that it does not recognize.

Table of contents and index: Allows you to automatically create a table of contents and index based on special codes that you insert in the document.

Thesaurus: A built-in thesaurus that allows you to search for synonyms without leaving the word processor.

Windows: Allows you to edit two or more documents at the same time. Each document appears in a separate window. This is particularly valuable when working on a large project that consists of several different files.

WYSIWYG (What You See Is What You Get): With WYSIWYG, a document appears on the display screen exactly as it will look when printed.

5.4. Summary:

At the end of this lesson you have learnt the importance of Microsoft Office products and important features of Word Processing. In all the MS-Office products, the familiar products are MS Word, MS Excel, MS PowerPoint and MS Access. Word processor enables you to create a document, store it electronically on a disk, display it on a screen, modify it by entering commands and characters from the keyboard and print it on a printer.

5.5. Technical Terms:

Cut and paste: Allows you to remove (cut) a section of text from one place in a document and insert (paste) it somewhere else.

Copy: Allows you to duplicate a section of text.

WYSIWYG: What You See Is What You Get.

Spell checker: A utility that allows you to check the spelling of words.

Word wrap: The word processor automatically moves to the next line when you have filled one line with text, and it will readjust text if you change the margins.

5.6. Model Questions:

1. Explain in detail about Ms-Office
2. Define Word-Processing? Explain the features of Word Processing?

5.7. References:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, ' The ABC'S Of Microsoft Office. ' BPB Publications, New Delhi

MS OFFICE 2000 Complete, BPB Publications, New Delhi

Y. SURESH BABU, M.Com., M.C.A.,

LESSON 6

MICROSOFT WORD

6.0 Objectives:

This lesson provides an introduction to the fundamentals of Microsoft Word. It covers starting Word, the Word interface, creating a Word document, editing a Word document, selection techniques, text formatting, page setup and print documents.

Structure of the Lesson:

- 6.1. Starting word and components of word window
- 6.2. Creating and saving a document
- 6.3. Adding & modifying text
- 6.4. Selecting Text
- 6.5. Cut, Copy and Paste Text
- 6.6. Paste Special
- 6.7. Find and Replace
- 6.8. Redo and Undo
- 6.9. Formatting Text & Paragraphs
- 6.10. Header and Footer
- 6.11. Bullets and Numbering
- 6.12. Zooming
- 6.13. Page Break
- 6.14. Auto Correct and Auto Text
- 6.15. Page Setup
- 6.16. Print Preview and Printing
- 6.17. Summary
- 6.18. Technical Terms
- 6.19. Model Questions
- 6.20. References

6.1. Getting Started with Microsoft Word:

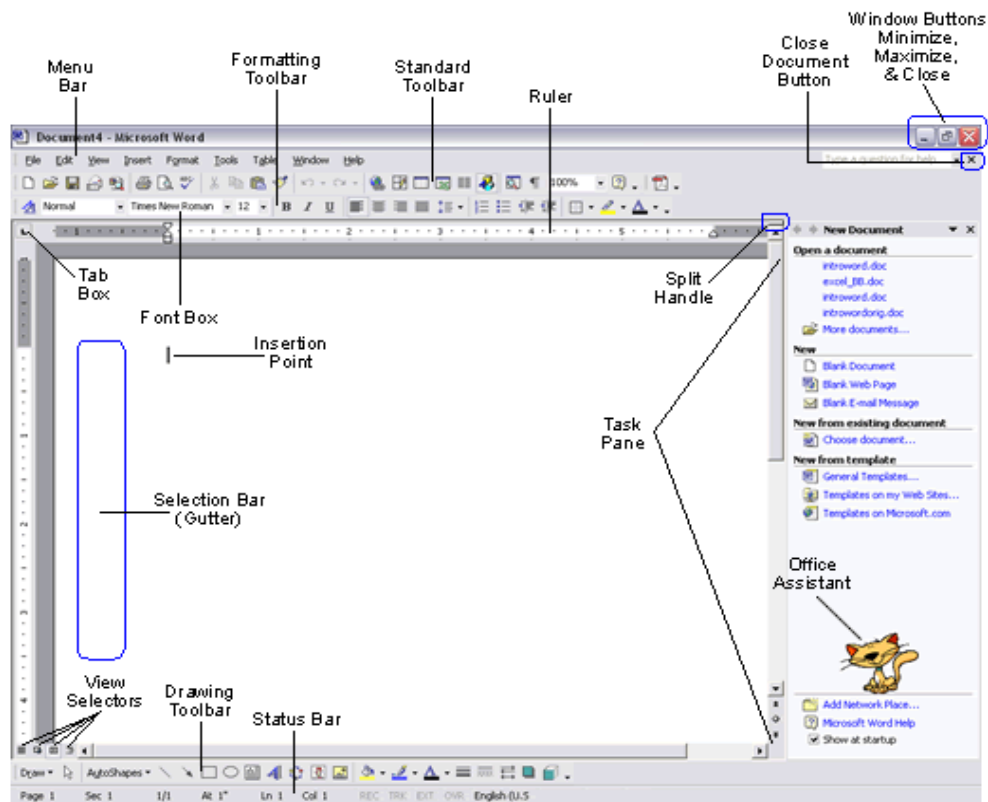
Starting Word

On the PC, select Start followed by Programs, and Microsoft Word from the Start list.

Components of the Word Window

Besides the usual PC window components (close box, title bar, scroll bars, etc.), a Word window has other elements, as shown in the following figure.

Figure 6.1



Component	Functionality or Purpose of the Component
Menu Bar	Contains File, Edit, View, Insert, Format, Tools, Table, Window and Help menus.
Standard Toolbar	Contains icons for shortcuts to menu commands.
Formatting Tool Bar	Contains pop-up menus for style, font, and font size; icons for boldface, italic, and underline; alignment icons; number and bullet list icons; indentation icons, the border icon, highlight, and font color icons.
Ruler	Ruler on which you can set tabs, paragraph alignment, and other formats.
Insertion Point	Blinking vertical bar that indicates where text you type will be inserted. Don't confuse the insertion point with the mouse I-beam. To move the insertion point, just click the mouse where you want the point moved.

End-of-File Marker	Non-printing symbol that marks the end of the file. You cannot insert text after this mark.
Selection Bar (Gutter)	Invisible narrow strip along the left edge of the window. Your mouse pointer changes to a right-pointing arrow when it is in this area. It is used to select a line, a paragraph, or the entire document.
Split Handle	Double-click to split the window in two (to view different portions of the same file). Double-click to return to one window
Status Bar	Displays page number, section number, and total number of pages, pointer position on page and time of day.
Task Pane	Displays and groups commonly used features for convenience.
Office Assistant	An animated character that can provide help and suggestions. There are multiple characters to choose from, and it is possible to turn the Office Assistant off.

6.2. Creating and Saving a Document:

Opening and Closing A Document

To open a file, select Open... from the File menu. In the dialog box, click on the filename and then on the Open button. If your file is not in the opened folder, click on the down arrow (next to folder name) to see a directory listing. The Up One Level button icon on the right enables you to move up one level. You can then switch to the floppy disk, hard drive or Desktop where the appropriate file or folder is located.

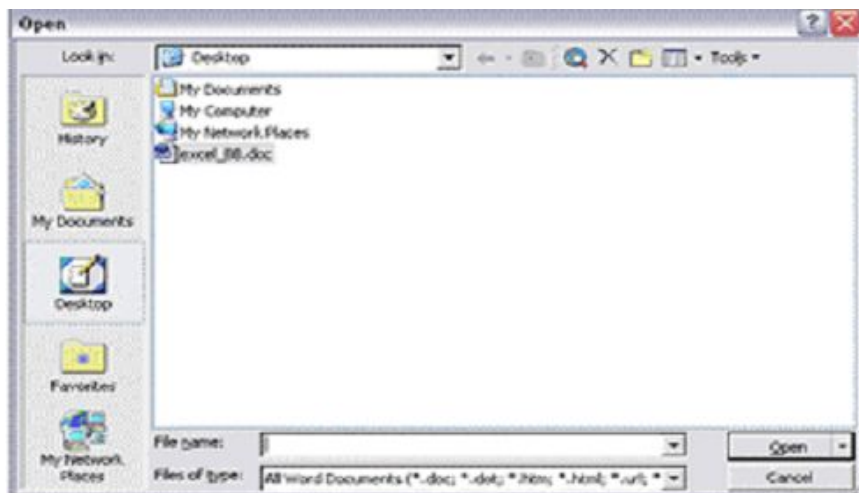


Figure 6.2

For example, to open a file from a floppy disk, select Open... from the File menu. If the disk's name does not appear on the left side of the dialog box, click on the down arrow. Select its name in the box at the left or if it is not displayed, click the Up One Level button to see a different display of files and folders. When you locate the desired diskette or file, click Open. (Double-clicking on the name will also open it).

To close a document, click its close box in the upper right of the title bar (box with the large X on it) or select Close from the File menu. The keyboard shortcut is <Ctrl + W>.

Saving a Document

Saving means preserve the content in a Secondary Storage device. It is a good idea to save your document right away and they continue saving every 5 or 10 minutes so if your computer crashes you will not be losing much. Your document and changes you make to a document are not saved to disk until you issue a save command. Saving is quick and easy. You should save often to minimize the loss of your work. Word has two save commands - Save and Save As. - that work similarly. Both are under the File menu.

Save	Save As
When you save a new document for the first time, Word displays a dialog box where you can choose a document name and where you want to save your document and disk. Use the Save As command whenever you give it a name. When you save an existing document that you have been editing, the newly saved version is written over the older version.	This command always displays a dialog box where you can choose a document name and disk. Use the Save As command whenever you want to save a copy of the current document under a different name or in a different folder (or disk). The newly saved copy becomes the active document.

6.3. Adding and Modifying Text

Entering Text

To enter text, just start typing. Word inserts the text you type at the insertion point. If you press a wrong key, use the **Delete** key to erase the mistake. Word automatically starts a new line when you reach the end of the current line. This is called "word wrap". Do not press Enter at the end of a line. Press Enter only at the end of paragraphs.

For typing a paragraph follow the procedure given below:

Remember, don't press Enter at the end of the line.

To move the insertion point with the mouse, click the mouse in the desired location. If the location you want is not visible, use the scroll bars to move up or down in the document. To move the insertion point to the top of the document, press <Ctrl + Home>. To move to the end of the document, press <Ctrl + End >.

To insert new text, just click the mouse at the point where you want to insert the additional text and start typing. Existing text moves to accommodate your additions.

Type **<Ctrl + Home>** to move to the very beginning of your document.

Deleting Text

The **Backspace** key deletes one character to the left of the insertion point. The **Delete** key deletes one character to the right of the insertion point. You can use these keys any time. To delete more than just a few characters, select the text and press the **Delete** key.

Undoing Mistakes

If you have accidentally deleted an important paragraph, use the **Undo** command under the **Edit menu** immediately (The keyboard shortcut is **< Ctrl + Z >**). The deleted portion of the text is brought back. This command can bring back the deleted contents only if you have not pressed any other key in between.

6.4. Selecting Text:

You select a portion of text in order to perform some operation on it. You can then delete, replace, copy, move, or format it. The Selection Bar is located to the left of the document window. When you put your mouse pointer, which is a left pointing arrow in that area, the arrow swings to the right. The following table gives the process of selection of text contents.

To Select...	Do this...
Word	Double click anywhere on the word.
Line	Click right-pointing arrow in the selection bar to left of line.
Sentence	<Ctrl + click> anywhere in sentence.
Paragraph	Double-click the right-pointing arrow in the selection bar to the left of the paragraph, or triple-click in paragraph.
A block of text of any size	Click at one end of the block and shift-click at the other end of the block. Or click at one end of block and drag to the other end of the block.
Entire document	Press < Ctrl + A > or hold down the Ctrl key and click in the selection bar.
Deselect text	Click anywhere outside the selection.
Extend a selection	Shift-click an additional block. For example, if you have selected a paragraph and you shift-click on another paragraph, it will be added to the selection.

6.5. Cut, Copy and Paste Text:

You can copy selected text from one place to another with the Copy and Paste commands.

Cut - The Cut command extracts the selected text and puts it on the Clipboard (a temporary storage area). The contents of the Clipboard are overwritten with each copy or cut. This can be done either by selecting Edit from the menu bar and opting the cut option in it. Alternative usage of the short cut key is CTRL +X.

Copy - The Copy command puts a copy of the selected text on the Clipboard. By selecting the Edit from the menu bar and opting the Copy option in it. Alternative is usage of the short cut key is CTRL + C.

Paste - The Paste command inserts the Clipboard's contents at the insertion point. The text that has been selected using Cut/ Copy command will be pasted at the insertion point. Existing text moves to accommodate pasted text. By selecting the Edit from the menu bar and opting the Paste option in it. Alternative usage of the short cut key is CTRL + V.

6.6. Paste Special:

Paste Special is a way to paste information from one application/document to another. The Paste Special dialog box allows you to paste information in different ways. Choose Paste Special from the Edit Menu.

You can paste in the information in several ways:

Formatted Text, Unformatted Text, Picture Object.

When we paste Formatted Text, it will paste all formatting (such as Bold, etc.) along with the text. When we use Paste Special option the formatted text can be pasted as unformatted text or an object.

6.7. Find and Replace:

Word allows us or helps us to search for a word in a document

Find: Find is used for searching a word in your document.

1. On the Edit menu, click Find.

2. In the **Find what** box, enter the text that you want to search for.

Replace: Replace is used to replace the searched word by another text.

Click On the Edit menu, click Replace.

In the **Find what** box, enter the text that you want to search for.

In the **Replace with** box, enter the replacement text.

The other options you have are Replace, Replace All, Find Next and Cancel.

Replace button replaces a word selected, at the first occurrence of the word in your document after the insertion point.

Replace All button replaces all the searched words in your document.

Find Next button searches for the next occurrence of the word being searched.

Cancel button cancels the search process.

6.8. Undo and Redo:

Undo

When working in Microsoft Word you will often commit mistakes. Instead of manually trying to undo these actions you are able to rectify the mistake just by clicking the Undo button on the Standard toolbar.

To practice the Undo Feature:

- 1) Press CTRL +End to move to the end of the document
- 2) Highlight the last paragraph in the document
- 3) Press the Delete key on the keyboard

To Undo the Previous Action:

Click the Undo button on the Standard Toolbar

Redo

You may accidentally click Undo too many times and thus, need to redo it. This command works exactly in the reverse of Undo command.

To Redo the Previous Action:

Click the Redo button on the Standard Toolbar

Save and close the file.

6.9. Formatting Text and Paragraph

Microsoft Word allows you to format a document very easily. Without formatting, a document looks very plain and may not get the attention that is desired.

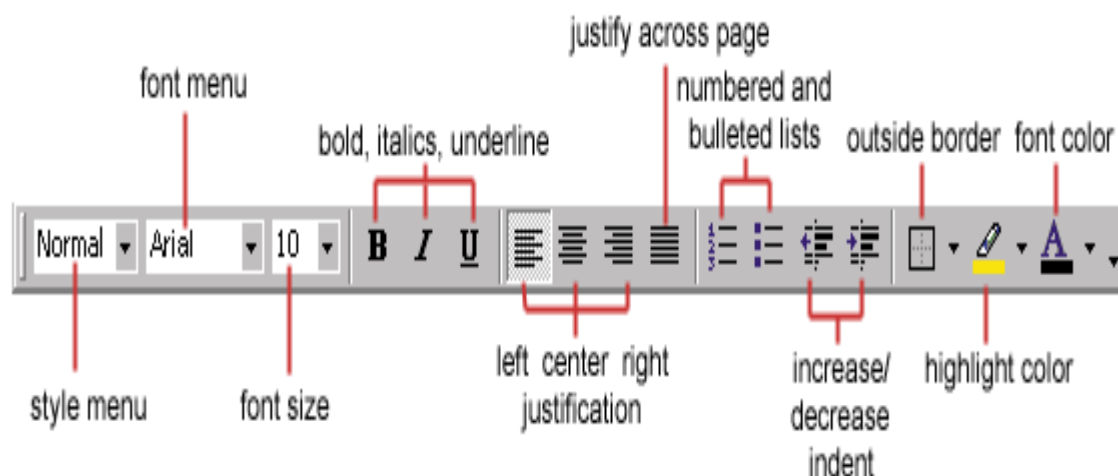


Figure 6.9a

Making Font Changes

In Microsoft Word you are able to change the font size, color, type and attributes. Below we demonstrate various ways to change the font.

Select a document of unformatted text for formatting.



Figure 6.9b

To edit the font style

- 1) Highlight the paragraph
- 2) Select Format menu
- 3) Select a Font type.
- 4) Notice the Preview window at the bottom of the dialog box, which displays the style of the font, selected.
- 5) Click OK

Font Size

Opting the one of the numbers in size box does changing the font size for a selected portion of the text.

Using Bold, Italics and Underline

Another way to make text stand out in a document is by using bold, italics and underline (or a combination of these) formatting.

To Bold Text in a Document

- 1) Select any word(s) in the document
- 2) Click the Bold icon on the Formatting Toolbar

To Italicize Text in a Document

- 1) Select any word(s) in the document
- 2) Click the Italic icon on the Formatting Toolbar

To Underline Text in a Document

- 1) Select any word(s) in the document
- 2) Click the Underline icon on the Formatting Toolbar

Formatting Paragraphs

In Microsoft Word, a paragraph is any amount of text followed by a paragraph symbol (¶). A one-word heading is a paragraph. A 100-word job description is a paragraph. Even a blank line terminated by a ¶ is a paragraph.

You can perform paragraph formatting from the Format menu.

Highlight the paragraph you want to format.

Click on Format from the Command menu.

Choose Paragraph.

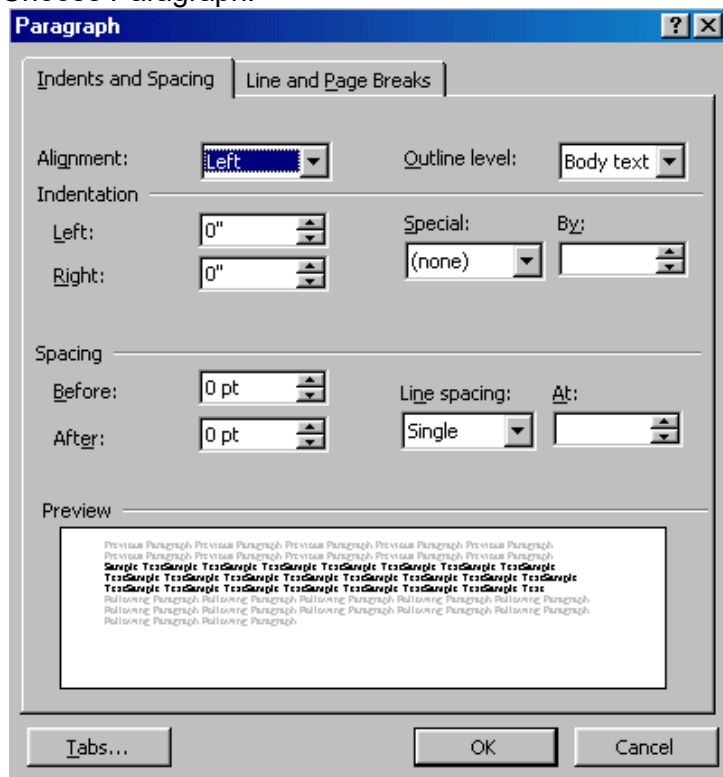


Figure 6.9c

Make the changes from the options that are displayed in the window. For example, to change line spacing from single to double, click on the Line spacing drop down arrow, and choose double. You can also choose options from the Formatting Tool Bar to format paragraphs.

6.10. Header and Footer

A header or footer is text or other information such as graphics that is stored at the top or bottom of the page throughout your document. You can use the same header and footer throughout a document or change for part of the document. For example, you can use your corporate logo in the first-page header, and then include the document's file name in the header for subsequent pages.

To view Headers and Footers in Normal View, click View > Header and Footer. If you are in Page Layout View (Word 97) or Print Layout View (Word 2000), simply double click the visible header or footer that appears as gray text.

In either case, the Header/Footer toolbar appears.

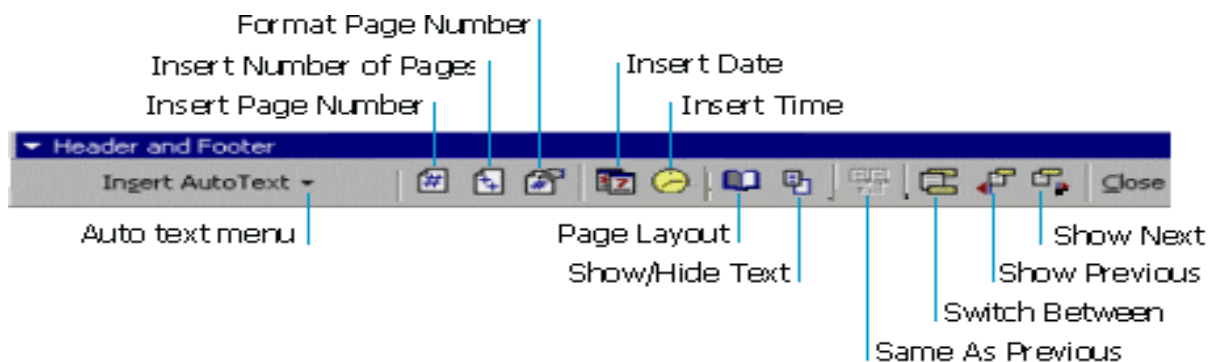


Figure 6.10

6.11. Bullets and Numbering:

Word has a feature of giving automatic numbers and bullets for paragraphs or for points. For bullets first select the paragraph or points to be bulleted and then select bullets icon from the standard toolbar or select Format from menu bar and opt for Bullets and numbering. Select different types of symbols as bullets.

6.12. Zooming:

The level of magnification of the text can be controlled by this zooming control. Choose percentage value from the zoom popup menu and size of magnification of text.

6.13. Page Break:

After allowing certain number of lines MS Word automatically inserts a dotted line in the document called page break. We can insert a page break at any line in the document. Select Insert Menu from menu bar and click break. Now the page break dialog box appears. Here select page break button and click on it. Then page break will be inserted at the current cursor position.

6.14. Auto Correct and Auto Text:

Auto Correct

Auto Correct is used to correct the repeated typing mistakes. Select Tools menu from the menu bar and click Auto Correct option, then activate Auto correct tab. Here type frequently misspelled words in 'Replace box' and the correct one's in 'with box'. Click the Add button to add this to the dictionary. Afterwards if we misspell a word, which was added to the dictionary, word will automatically correct the mistake.

Auto Text

Frequently typed text can be placed into an Auto Text entry. First type the text what we want to put into an Auto Text entry and then select it. Select Tools menu and Click Auto Correct and then activate Auto Text Tab. Now, enter the abbreviation for the selected text and click OK. Afterwards, if we want to type the text that is there in the Auto Text, simply type the abbreviation and click OK. Then automatically the text will be typed without actually typing.

6.15. Page Setup:

Page Margins

The page margins of the document can be changed using the rulers on the page and the Page Setup window. The ruler method is discussed,



Figure 6.15a

first move the mouse over the area where the white ruler changes to gray.

When the cursor becomes a double-ended arrow, click with the mouse and drag the margin indicator to the desired location. Release the mouse when the margin is set.

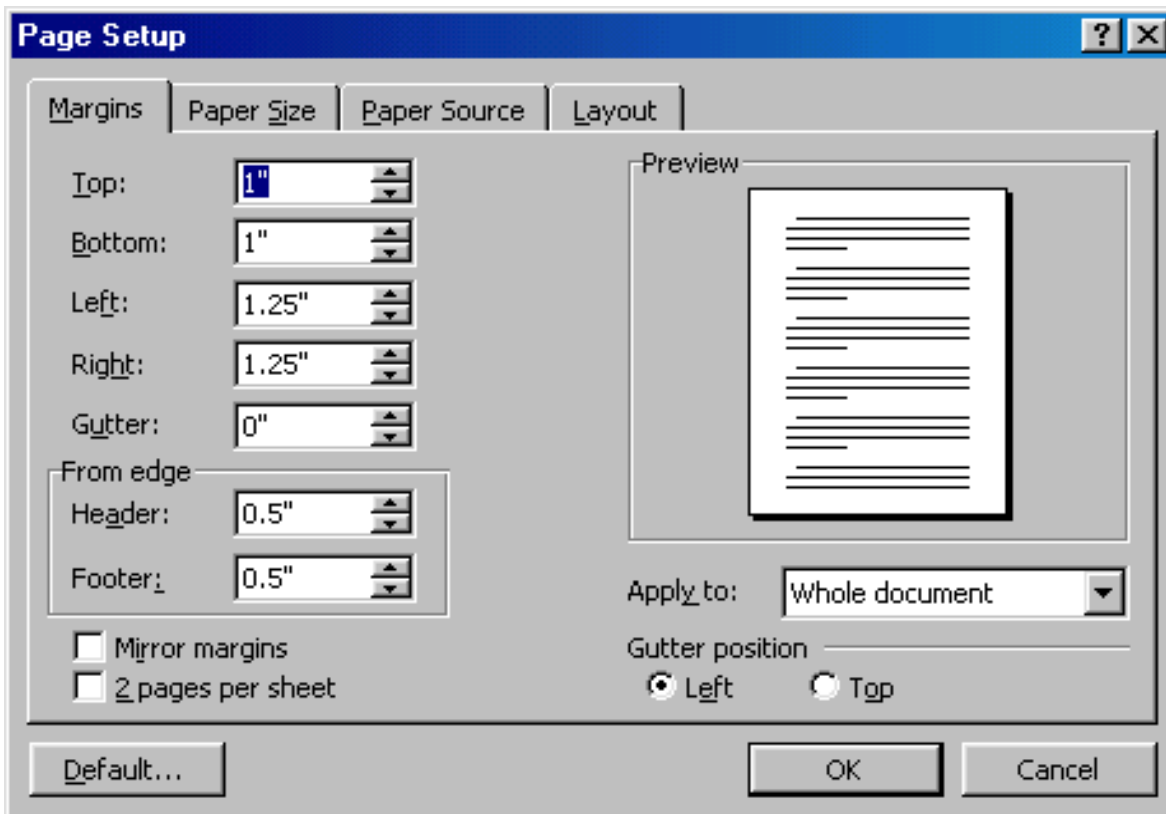


Figure 6.15b

The margins can also be changed using the Page Setup dialog box: Select File/Page Setup and choose the Margins tab in the dialog box.

Enter margin values in the Top, Bottom, Left, and Right boxes. The Preview window will reflect the changes. If the document has Headers and/or Footers, the distance this text appears from the edge of the page can be changed. Click OK when finished.

Page Size and Orientation

Change the orientation page within the Page Setup dialog box. Select File/Page Setup and choose the Paper Size tab.

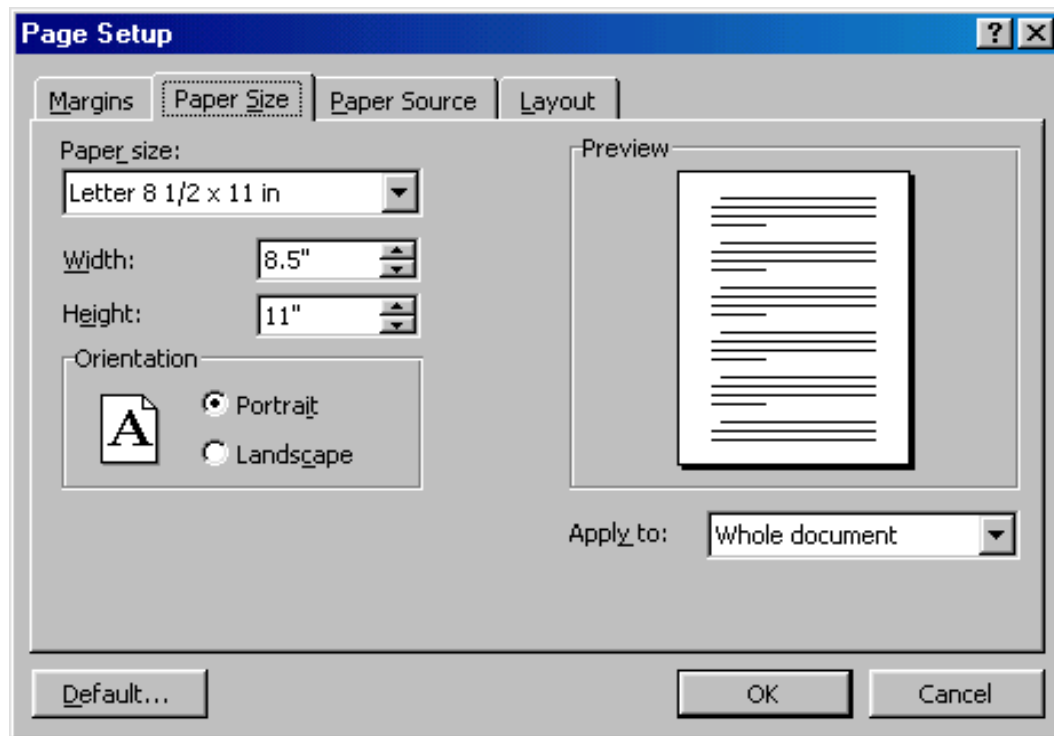


Figure 6.15c

Select the proper paper size from the drop-down menu.
Change the orientation from Portrait or Landscape by checking the corresponding radio button.

Page Numbers

Follow these instructions to add page numbers to a document.
Select Insert/Page numbers from the menu bar and the following dialog box will appear.

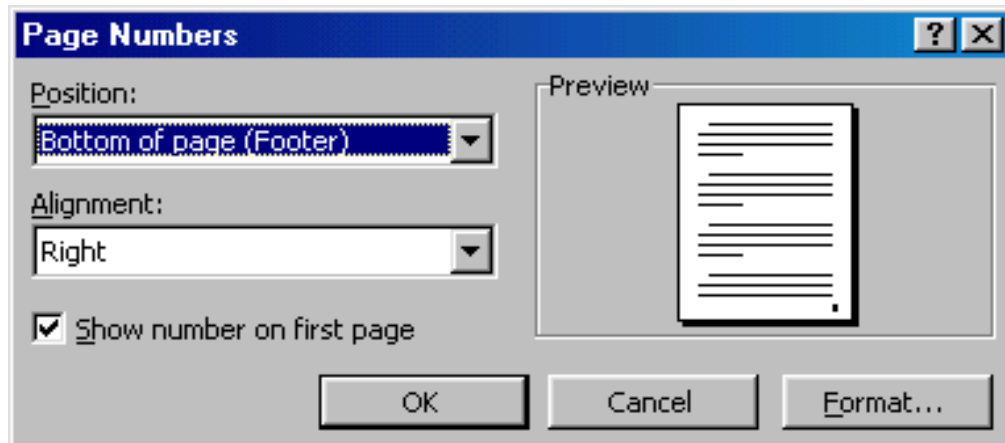


Figure 6.15d

Select the position of the page numbers by choosing "Top of page" or "Bottom of page" from the Position drop-down menu. Select the alignment of the page numbers in the Alignment drop-down menu. If you do not want the page number to show on the first page (if it is a title page, for example), uncheck the Show number of first page box. Click OK when finished.

6.16. Print Preview and Printing:

Preview your document by clicking the Print Preview button on the standard toolbar or by selecting File/Print Preview. When the document is ready to print, click the Print button from the Print Preview screen or select File|Print.

6.17. Summary:

At the end of this lesson you have learnt the important features of Ms-Word. We started with procedure to invoke Ms-Word. We have identified toolbars available in the word. You learnt the procedure of creating a new document and opening an existing document. Now you are in a position to create a new document and can make corrections like deleting a character, word, line or paragraph and replace with required text by using replace option. You can also move the selected part of the text to another place by the "cut" and "paste" commands. At the end of the lesson we discussed the procedure of saving a document and exit from word.

You have gained the knowledge of formatting a document using different formatting options like setting margins, changing the font size, and alignment to the text. The headers, footers, tabs, indents make the document in presentable format.

6.18. Technical Terms:

Font

Font refers to the combination of a **typeface** (eg. Times Roman, Arial, Century Schoolbook) and its **attributes** (bold, italics, size (12pts)).

Bullets

Bullets are special characters or symbols that are used to set off a paragraph. Typically, each bulleted paragraph is an item in a list, and it consists of the bullet symbol and indenting to set the bullet apart from the text.

Document

A document is any data file that is created by a specific application.

Current Document

The current, or active document is the document you see on the Word screen.

Drag-and-drop

Drag-and-drop is a feature that allows you to move or copy information without using the Windows Clipboard. To use it, you simply drag a selected item from one location to another. It is best used for moving or copying small items short distances.

Footer

A Footer is the contents of an area located within the bottom margin of a page.

Header

A header is the contents of an area located within the top margin of a page.

Justification

Justification determines how lines and characters within those lines are printed. With full justification, all lines start at the left margin and end at the right margin.

Margin

The margin is the amount of blank space, usually measured in inches or characters, above and below and to the right and left of the main body of a document.

Menu Bar

The menu bar contains the names of Word's menus and is used to navigate through their commands.

Paragraph

A paragraph in Word begins where you start typing, and it ends where you press [ENTER].

6.19. Model Questions:

1. Explain about different components of MS-Word window.
2. Explain Find, Replace & Go To?
3. Explain Header and Footer?
4. What is the difference between Auto Correct and Auto Text
5. Explain the text formatting features in MS-Word

6.20. References:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, ' The ABC'S Of Microsoft Office. ' BPB Publications, New Delhi

MS OFFICE 2000 Complete, BPB Publications, New Delhi

Y. SURESH BABU, M.Com., M.C.A.,
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson 7

MICROSOFT WORD

7.0 Objectives:

This lesson provides an introduction to the Views, Spell Checking, Tables, Macros and Mail merge.

Structure of the Lesson:

- 7.1. Views
- 7.2. Spell Checker
- 7.3. Word Count
- 7.4. Macro
- 7.5. Graphics
- 7.6. Lines and Shapes
- 7.7. Word art
- 7.8. Tables
- 7.9. Borders and Shading
- 7.10. Footnote and Endnote
- 7.11. Tabs
- 7.12. Mail merge
- 7.13. Format Painter
- 7.14. Summary
- 7.15. Technical Terms
- 7.16. Model Questions
- 7.17. References

7.1. Views:

Any document can be viewed in different fashions. These views can be selected from the view menu of the menu bar.

Normal view is a default view for all new documents. However saved document will be opened last saved view. For example a document is opened in normal view and saved in page layout view, when saved document is reopened, the document will be opened in page layout view.

Normal View

You can create and edit text or graphics normally. To work with header and footers you invoked select item/Header and Footer. Footer details will not be displayed on screen.

This view will not show side-by-side column positioning, footers and headers. Automatic page breaks are shown as dotted lines. Manual page breaks containing the words " Page Break".

Page layout View

In this view headers & footers can be seen and edited. You can modify the Header/Footer by the selecting the matter itself. It will display the left side scale also.

Outline View

This view is used to see entire contents of the document, chapter headlines and sections headings if your document is properly formatted. It is useful if the document is large.

Print Preview

Print preview is another way to view a document. Choose file → print preview or Ctrl +F2. You can see one page to six pages at a time in print preview.

Split Screen View

You can see two parts of your document at once. This feature is useful to make copying, cutting and pasting quickly in both the documents simultaneously.

7.2. Spelling Checker:

To start the spelling checker, select Spelling and Grammar from the Tools menu. MS Word will begin to check for misspellings. If a word is found that is not in its dictionary, it pauses, displays the word, and if possible, suggests alternatives (see figure below).

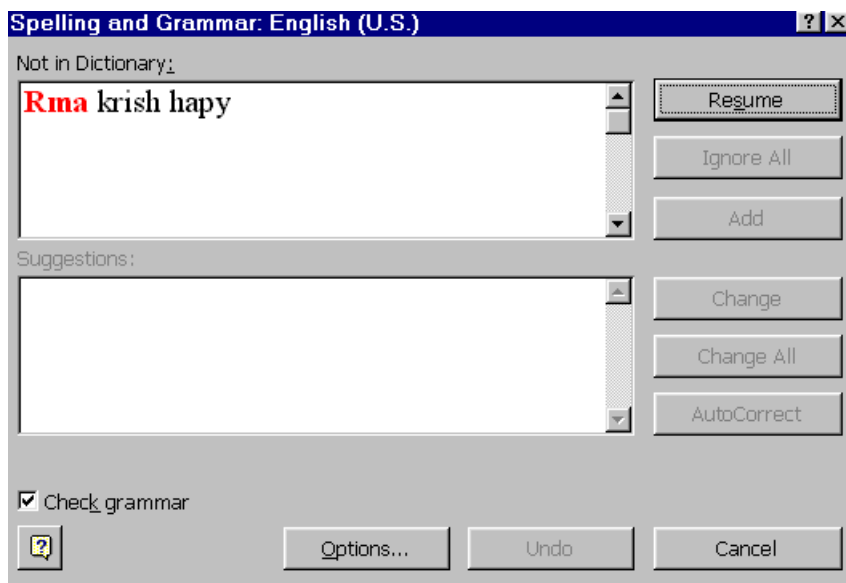


Figure 7.2

You have several choices at this point described in the table below. If the word is spelled correctly, you can Ignore it or add it to a custom dictionary by selecting the options. If the word is incorrect, you can use Replace it with one of its suggestions or with a spelling you type in the Change To: box.

If the word is spelled correctly...	
To ignore only this occurrence of the word, click	Ignore
To ignore all occurrences of the word, click	Ignore All
To add the word to a custom dictionary, click	Add
If the word is misspelled...	
First, click correct spelling in Suggestions list or type the correction in the Change To: box. Then...	
To change only this occurrence of the word, click	Change
To change all occurrence of the word, click	Change All

7.3. Word Count:

Word count displays the number of words, paragraphs, lines, characters, pages in the document. Select Tools menu from the menu bar and click on the word count.

7.4. Macro:

Macros are advanced features that can speed up editing or formatting you may perform often in a Word document. They record sequences of menu selections that you choose so that a series of actions can be completed in one step.

Recording A Macro

To record a macro, follow these steps:

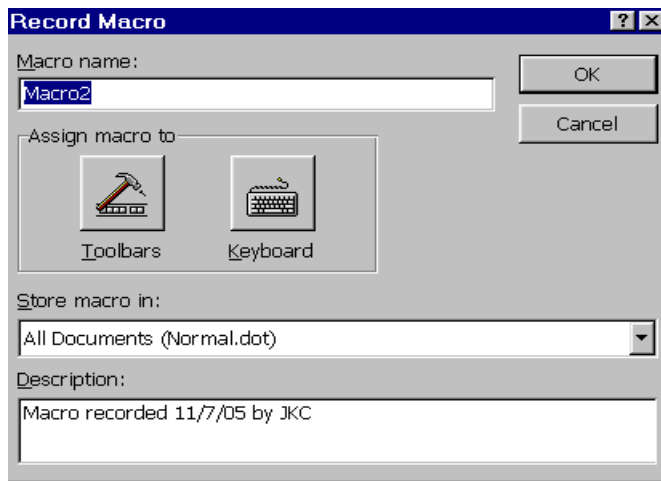


Figure 7.4a

Click Tools/Macro/Record New Macro on the menu bar.

Name the macro in the Macro name field. This name cannot contain spaces and or begin with a number.

From the Store macro in drop-down box, select the document you would like the macro to be associated with or choose "All Documents" enabling to use the macro in any document. Enter a description of the macro in the Description field. This is for your reference only so you remember what the macro does. Click OK to begin recording.

Select options from the drop-down menus and Word will record the options you choose from the dialog boxes, such as changing the margins on the Page Setup window. Select only options that modify the document. Word will not record toggle actions such as View/Toolbars that have no effect on the document itself.

The recording toolbar will allow you to stop, pause, and resume recording. Click the Stop button in the recording toolbar. The macro is now saved.



Figure 7.4b

Running A Macro

To run an existing macro, follow these steps:

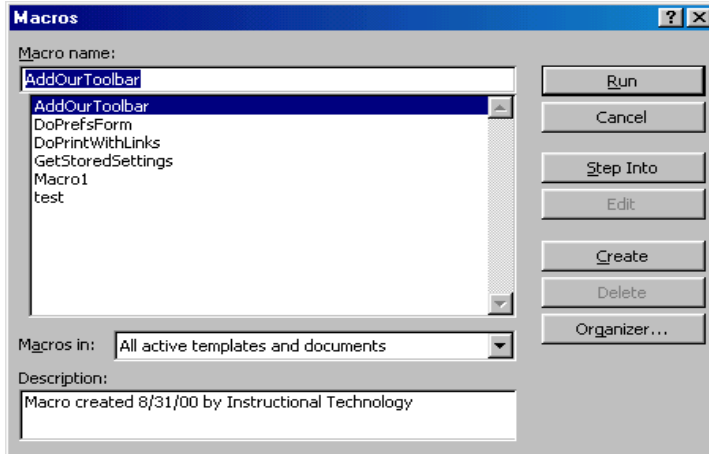


Figure 7.4c

Select Tools/Macro/Macros from the menu bar.

From the Macros window, highlight the Macro name in the list and click Run. If the macro is long and you want to stop it while it is running, press BREAK (hold CTRL and press PAUSE).

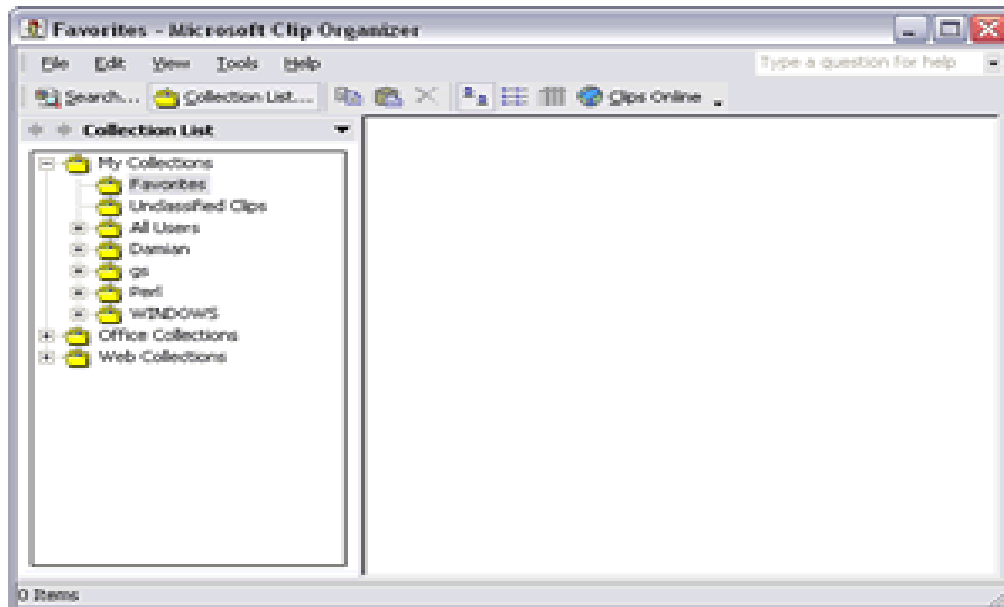


Figure 7.5.

7.5.Graphics:

Microsoft Word contains a full set of graphic tools. You can:

Create graphics from scratch.

Import graphics from other programs.

Place graphics within text.

Manipulate graphics on a page.

Word comes with some pre-designed graphics that are normally installed in the MS OFFICE\CLIP ART folder. These graphics can be resized and placed within text.

Inserting a Graphic in your Document

To insert a graphic in a document position the insert point where you want to place the graphic in the text. Select Picture from the insert menu. Click on Clip art, and double-click the graphic you want to insert.

Choose Insert, Picture, Clip Art or click the Insert Clip Art button on the Drawing toolbar.

Word opens the Clip Art Task Pane.

Search for a specific clip image by entering what you are looking for within the Search field in the Task Pane.

Alternately, click on the Clip Organizer button toward the bottom of the Task Pane to open up the Clip Art Organizer.

Select a clip and when Word shows a control palette, select the first option, Insert Clip. The clip is inserted where your insertion point is positioned.

You can also drag the clip art into your document.

Wrapping Text around a Graphic

By default text does not wrap around a graphic. To get text to wrap around a graphic, do the following:

Right click on the graphic to get the short cut menu.

Choose Format Picture from the list displayed.

Select the Layout tab.

Choose the style of wrap that you like and click OK.

Resizing a Picture

Select the inserted picture by clicking once on it.

Click the Cropping tool on the Picture toolbar.

Notice that the mouse cursor has changed.

Drag one of the corners with the pointer to reduce its size and eliminate it.

7.6. Lines and Shapes

Use the Drawing toolbar to create lines, shapes, text boxes, AutoShapes, WordArt, or Clip Art.

Drawing a Line or Arrow

Open a new Word document. Click the Drawing tool in the Standard toolbar or right click in any toolbar on your screen to display a list of available toolbars. Select the Drawing toolbar. Click a line or arrow on the Drawing toolbar. When the cross hair (+) appears, point to a spot in your document where you want the line or shape and draw a shape.

Drawing an AutoShape

Click AutoShapes on the Drawing toolbar. Select Stars and Banners from the drop down list of categories. Select the shape you want. Click in your document and drag the cross hair that appears to draw the shape.

Adding a Text Box:

Click the Text Box icon on the Drawing toolbar. When the cross hair appears, point to a spot in your document where you want the text box and drag. The box appears with an insertion point in it. Type "Microsoft Word Graphics" within the box.

7.7. WordArt :

To display the Word Art Gallery, choose Insert, Picture, WordArt or click the WordArt icon on the Drawing toolbar. Double-Click a WordArt style. Word opens the Edit WordArt dialog box. Type the text you want and it will replace the sample text. Choose a font, font size, and style. Click OK to close the WordArt Gallery and see your text.



Figure 7.7

7.8. Tables:

Creating Tables Using the Menu

A table is a grid of rows and columns on the page, like a spreadsheet. The area where a row and column intersect is called a cell. The number and width of columns can vary from row to row. Cells are separated by gridlines, which are non-printing separators. To display the table gridlines, select Gridlines from the Table menu.

Each cell can contain a paragraph, multiple paragraphs of text, or graphics. Microsoft Word wraps text within a cell in the same way that text is wrapped at the end of a line of normal text. Text within cells can be formatted, copied, cut, and pasted. You can add text, edit text, change its size, add borders, or add and delete columns and rows.

To create a table, select the Insert Table command from the Table menu. The Table Dialog box appears: enter the number of columns and rows you want to include in the table and click on the OK button. A table with the number of columns and rows that you specified is inserted in your document and the cursor is positioned in the first cell of the table. You can move from cell to cell using the <Tab> key, to move forward to the next cell and the <Shift Tab> keys to move back to the previous cell.

Creating a Table Using the Table Button in the Standard Toolbar

The Table button on the Toolbar provides another way to create a table. To use the Table button, place the mouse on the Table button and press the mouse button. The Table button selection box pops down. Drag the mouse down to select the number of rows you want and to the right to select the number of columns. When you have selected the correct number of rows and columns, release the mouse button. A table with the number of cells you have specified is inserted in your document.

Selecting Tables

Cells in a table can be selected using the click-and-drag method or one of the following techniques:

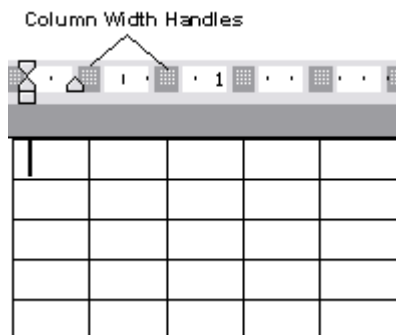
To select a cell:	Click in the lower left corner of the cell. (The mouse pointer shape becomes a right pointing arrow.)
To select a row:	Single Click in the selection bar to the left of the row.
To select a column:	Place the mouse on the top border of the column and click. (The mouse pointer becomes a black down pointing arrow.)
To select the entire table:	On a Macintosh hold down the Option button and double-click in the table. On a Windows machine Press the <Alt> key and the number 5 key on the numeric keypad . (The NumLock must be turned off.)
To select more than one cell:	Drag the mouse across the desired cell selection.

Inserting Columns and Rows

To insert a row or column within a table, select Insert Row or Column options from the Table menu and position the insertion point in the row/column you want to insert.

Changing Column Width

You can adjust the width of a column, or you can adjust the width of all the cells in a column. To change the column width, click on the column width button. Drag the button to the left or right according to your needs. To change the width of all the columns select the entire table and drag the column width button you wish to change.



Click and drag the first column width button to the 1.5-inch mark. Click on the Next Column width button to the 3-inch mark. The mouse can also be used to change column widths. To select the column that you want to change, place the mouse pointer on the right border of the column. As you

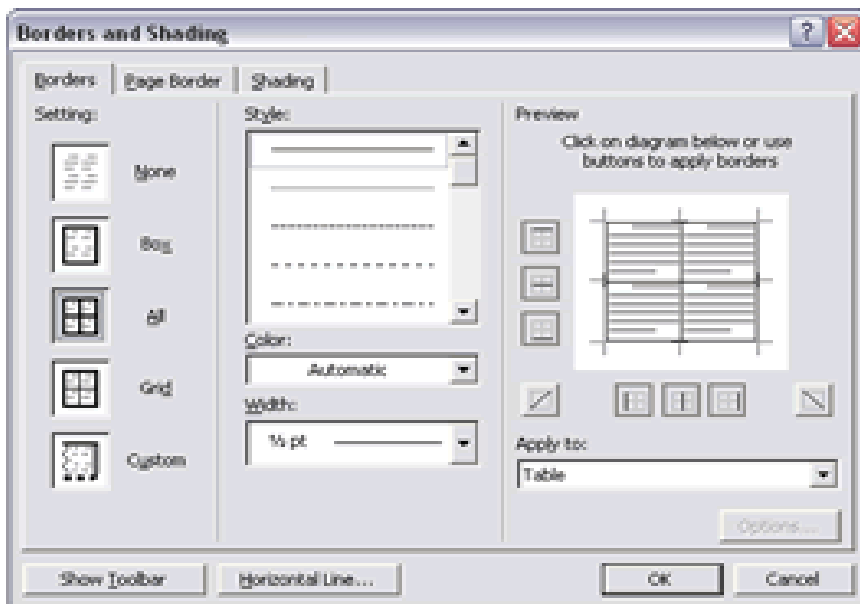
move the mouse pointer on top of the column border the pointer becomes a two-headed arrow (see example). When the mouse pointer is displayed as the two-headed arrow, press the mouse button and drag to reposition the column width. As you drag the mouse, an outline of the new border shows the border position. When you have resized the column to a satisfactory width, release the mouse button. The entire column is resized accordingly.

Deleting Columns and Rows

To delete a column or row, select the column or row you want to delete then select Delete column or row from the Table menu. Notice that if you select a column, the command is Delete Columns. If you select a row, the command is Delete Rows.

7.9. Borders and Shading:

You can add a border to any or all sides of each page in a document, to pages in a section, to the first page only, or to all pages except the first. You can add page borders in many line styles and colors, as well as a variety of graphical borders.



7.10. Footnote and Endnote:

Footnotes and Endnotes are used in printed documents to explain, comment on, or provide references for text in a document. You can include both Footnotes and Endnotes in the same document — for example, you might use Footnotes for detailed comments and Endnotes for citation of sources. Footnotes appear at the end of each page in a document. Endnotes typically appear at the end of a document.

1. In print layout view, click where you want to insert the note reference mark.
2. On the **Insert** menu, click **Footnote**.
3. Click **Footnote** or **Endnote**.
4. Under **Numbering**, click the option you want.
5. Click **OK**.

Word inserts the note number and places the insertion point next to the note number.

6. Type the note text.
7. Scroll to your place in the document.

7.11. Tabs:

By using tab key on the keyboard we can move the cursor by a fixed distance in the document. By default, every tab press, the cursor moves half-inch. But we can create our own tab positions. For creating customized tab positions first we have to select the tab alignment button at the left hand side corner of the ruler line. Tabs can also be created, by using the tab dialog box. Select Format menu from the menu bar and click tabs option. Then we get Tabs dialog box. Here we select the type of the tab and distance and click OK. button. Then the tabs will be set accordingly. Here by using tab dialog box we can also create one more tab i.e., bar tab. By this we can create a vertical line in the text at the specified bar tab position.

7.12. Mail Merge:

Mail Merge is used to quickly create form letters, mailing labels, envelopes and catalogs by merging the information from two different files. Those two files are generally the Main Document and Data Source File. After creating those two files, they are merged to get the desired result. Generally mail merge is used to print multiple copies of a particular letter addressed to many members. For example Circulars, Greetings & Invitations will come under mail merging.

Main Document

The main document contains the subject matter of the letter to be printed. It can contain text, graphics objects and fields (data), which are to be inserted in to the letter. When merging, data will be merged with the main document.

Data Source

Data Source file contains the information in the form of records, which is to be inserted to the main document. In one record, one member's information is stored – for example one person's address.

The individual information in the records is called a field. Actually the data written in these fields will be inserted in the main document during merge.

Merge requires 3 steps

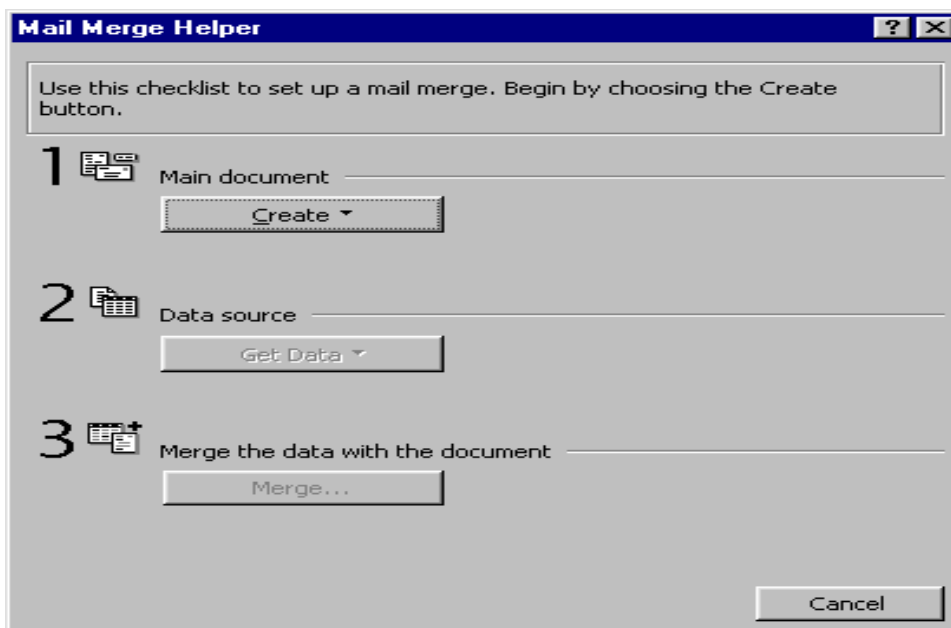
Creating Main Document

Creating Data Source

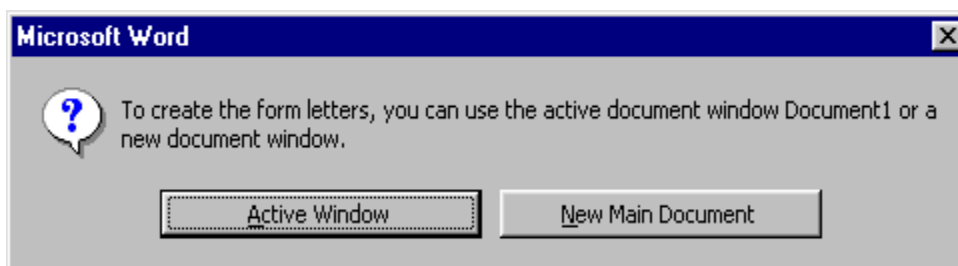
Merging two files.

Creating Main Document

First, Open a new document to create a main document by selecting File→New. Then select Tools→Mail Merge. The following dialog box will be displayed on screen:



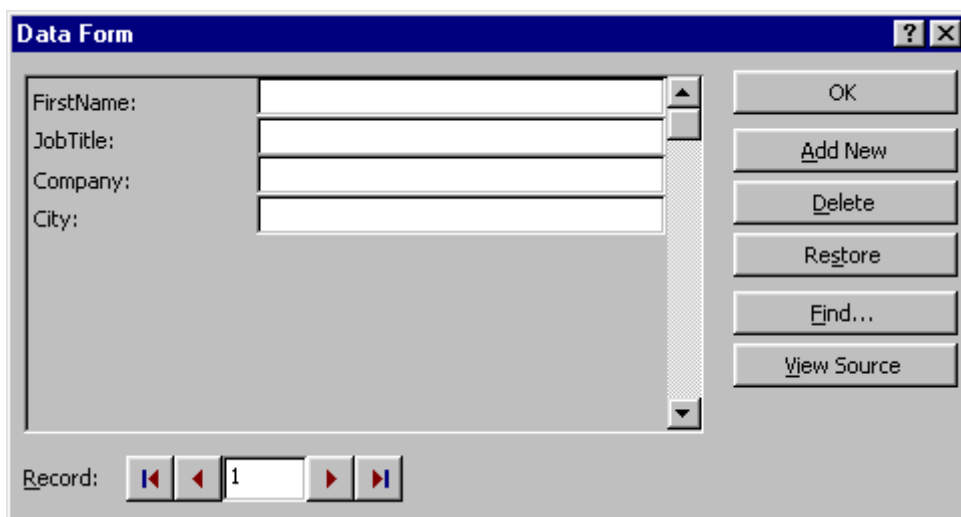
Click on the “Create” button to display the available formats (form letter, mailing labels, envelopes or catalog) of the main document. Then a dialog will be displayed prompting where to create the main document (whether the active window or a new document). Click on “active window” to create the main document in the active window.



Creating Data Source

To create a data source file click on Get Data in Data Source. The following will be displayed. Click on Create Data Source to specify the fields to be included in the address, like name, place etc.,

The existing field names will be displayed on the screen to the right. Unwanted fields can be removed by clicking on the field name and by clicking on the field name and by clicking on the button “remove field name”. If any new fields are to be added, type the field name on the left side. After specifying the field names, to type the contents, click on ok. A dialog box will be displayed to specify a name for the data source. Type the name and click on “Save”. A dialog box is displayed asking to edit the data source or edit the master document. The following diagram specifies the same:



Type the details by pressing enter after typing the first name, job title etc., for this, click on “Add new”. After typing all the details click on OK. Add contents to the document. Insert the merge fields in the window document. Finally select tools→mail merge. Merge dialog box will again get displayed, click on “Merge”.

7.13. Format Painter:

The **Format Painter** is a button on the Standard toolbar. It makes it easy to copy all the formatting features of one paragraph to one or more paragraphs.

To copy the paragraph formatting using the Format Painter you will:

1. Highlight the paragraph whose format you want to copy.
2. Now click the Format Painter button on the Standard toolbar.
 - If you press the button once, you will be allowed to copy the format to only one paragraph.

- If you double-click the button, the button will stay depressed and allow you to copy the format to as many paragraphs as you want. When you are finished you must then click the Format Button once in order to turn off the feature.
3. Notice that your insertion point now has a Paint Brush attached to it. This indicates that the format of the selected paragraph may be painted or copied onto another paragraph.
 4. Place your insertion point anywhere within the paragraph that is to receive the format and click the left mouse button. At this point the format of this paragraph will change.

7.14. Summary:

In this lesson we discussed the procedure of inserting a table in the document. You can insert a table of required number of rows and columns. In this lesson you have learnt the concept of mail merge and its usefulness. The mail merge feature supports many word processors that enable you to generate form letters. The mail merge feature of Ms-Word requires mainly two components: the main document and the data file. The main document is the body of the letter that has to be sent. The data file contains the list of names and addresses in the form of rows. You have learnt the procedures of creating the main document and the data file, and combining them.

You have observed how to use spell checker, the macro capability, and working with the graphics like ClipArt, WordArt, Format Painter, Footnote and Endnote features.

7.15. Technical Terms:

ClipArt

ClipArt consists of pre-designed images that can be placed within a document.

Data Form

A data form is a dialog box that allows you to view and edit individual records in a database.

Data Source

A data source contains the information from which a merged document is created. The data source is merged with a main document, which specifies the kind of output required.

Database

A database is a collection of organized information.

Dictionary

Word uses two kinds of dictionaries, a main dictionary and a custom dictionary. The main dictionary cannot be altered; we can add words to the custom dictionary.

Mail Merge

Mail Merge refers to the process of combining a data source, and a main document to produce a unique output.

Wizard

A wizard is a feature of Microsoft applications that lets you work through a series of dialog boxes to help you complete a task.

7.16. Model Questions:

1. What is a macro? Explain the procedure for creating a macro?
2. How do you perform mail merge?
3. Explain the difference between Footnote and Endnote?
4. Explain the procedure to create a table?
5. Explain different views in Ms Word ?
6. Explain the graphic facilities in MS-Word ?

7.17.Refereneces:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, ' The ABC'S Of Microsoft Office. ' **BPB Publications, New Delhi**

MS OFFICE 2000 Complete, BPB Publications, New Delhi

Y. SURESH BABU, M.Com., M.C.A.,
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson 8

MICROSOFT POWERPOINT

8.0 Objective:

This lesson introduces the basic features of Microsoft PowerPoint, Creating Presentations, View options, Inserting new slides and setting background color scheme to the presentation.

Structure of the Lesson:

- 8.1. PowerPoint basics
- 8.2. Starting a presentation
- 8.3. PowerPoint window properties
- 8.4. Save a PowerPoint presentation
- 8.5. Copy and Move data in a PowerPoint presentation
- 8.6. View options
- 8.7. Add a slide
- 8.8. Add/Modify text
- 8.9. Bullet styles
- 8.10. Speaker notes
- 8.11. Background/Color scheme
- 8.12. Slide master
- 8.13. Headers/Footers
- 8.14. Summary
- 8.15. Technical Terms
- 8.16. Model Questions
- 8.17. References

8.1. PowerPoint Basics:

PowerPoint is an application that lets you build, print, and deliver presentations. You have several options for the delivery of a presentation, but you only have to develop it once. You can print slides, print handouts, print notes pages, prepare for 35mm slides, or deliver an on-screen presentation. Only the on-screen presentation allows you to use the full range of PowerPoint's features.

As you develop a PowerPoint presentation, it is important to remember that you should not try to include every piece of information you wish to deliver. PowerPoint slides should contain brief, concise, descriptive phrases that will help you remember what you want to present and to serve as a reminder for your audience.

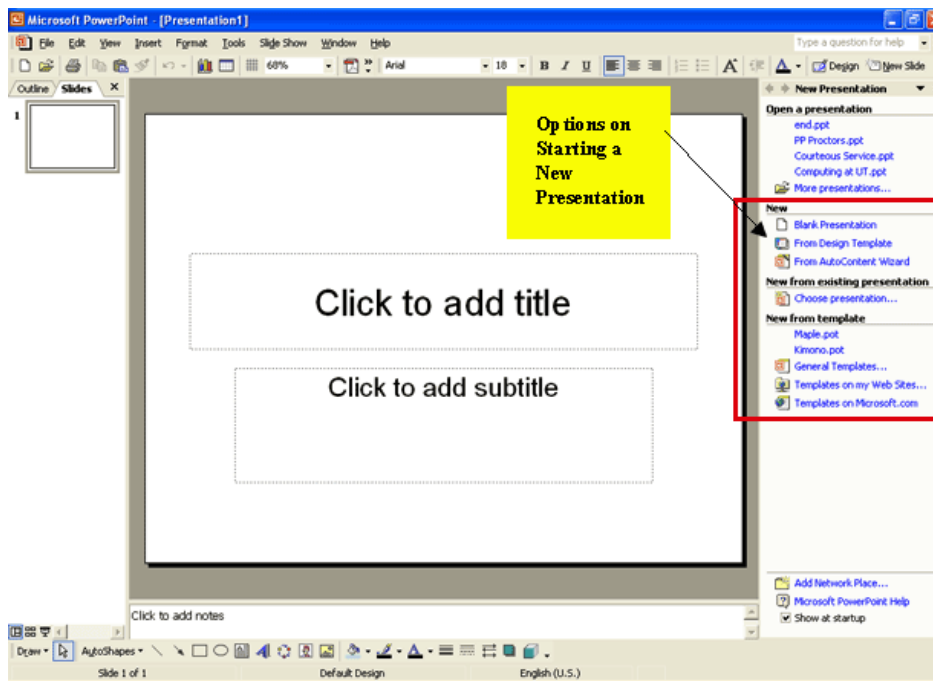
8.2. Starting A Presentation:

Start PowerPoint by either of two methods:

1. Go to the Start menu. Choose Programs and then click on Microsoft PowerPoint.
2. Double-click the icon of any PowerPoint document. When you double-click a PowerPoint document, PowerPoint opens with the document already loaded.

A PowerPoint presentation consists of slides that can contain text, graphics, charts, and other data types. When you start PowerPoint, you can start with a blank presentation, or you can begin from a template or use the AutoContent Wizard.

The AutoContent Wizard is series of step-by-step instructions designed to assist you. The Template button to help accesses slide templates create a consistent, professional look for your slide resenation.



New Presentation Dialog Box

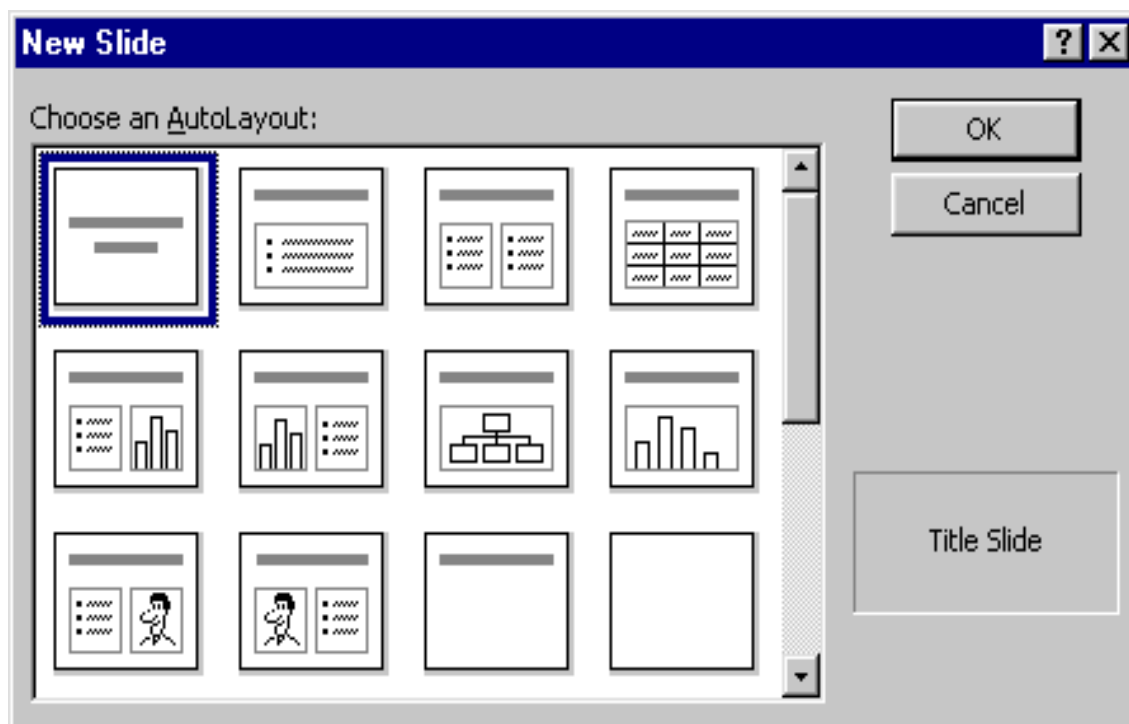
AutoContent Wizard walks you through a series of questions designed to elicit information about you and the type of presentation you are making. The Wizard then builds a dummy presentation that will guide you in developing the content of the presentation.

Design Template allows you to establish the background and color scheme from the available templates prior to beginning work in the new presentation. Template choices are generally easier to make after you have opened the new presentation because you can better see the characteristics of each template before making a selection.

Blank presentation opens a new presentation with no template. Open an existing presentation displays a list of recently opened PowerPoint presentations from which you can choose. Or, you may choose More Files to move to the location of a previously saved presentation file and open it.

AutoLayout

After you have opened a new presentation, PowerPoint displays the New Slide dialog box containing several Auto Layouts. Auto Layouts provide a pre-determined layout for each specific type of slide. They provide consistency throughout the presentation. Each layout depicted is described in the lower right corner when you click the layout. This sample New Slide dialog box shows the Title Slide selected (denoted with the thick border).

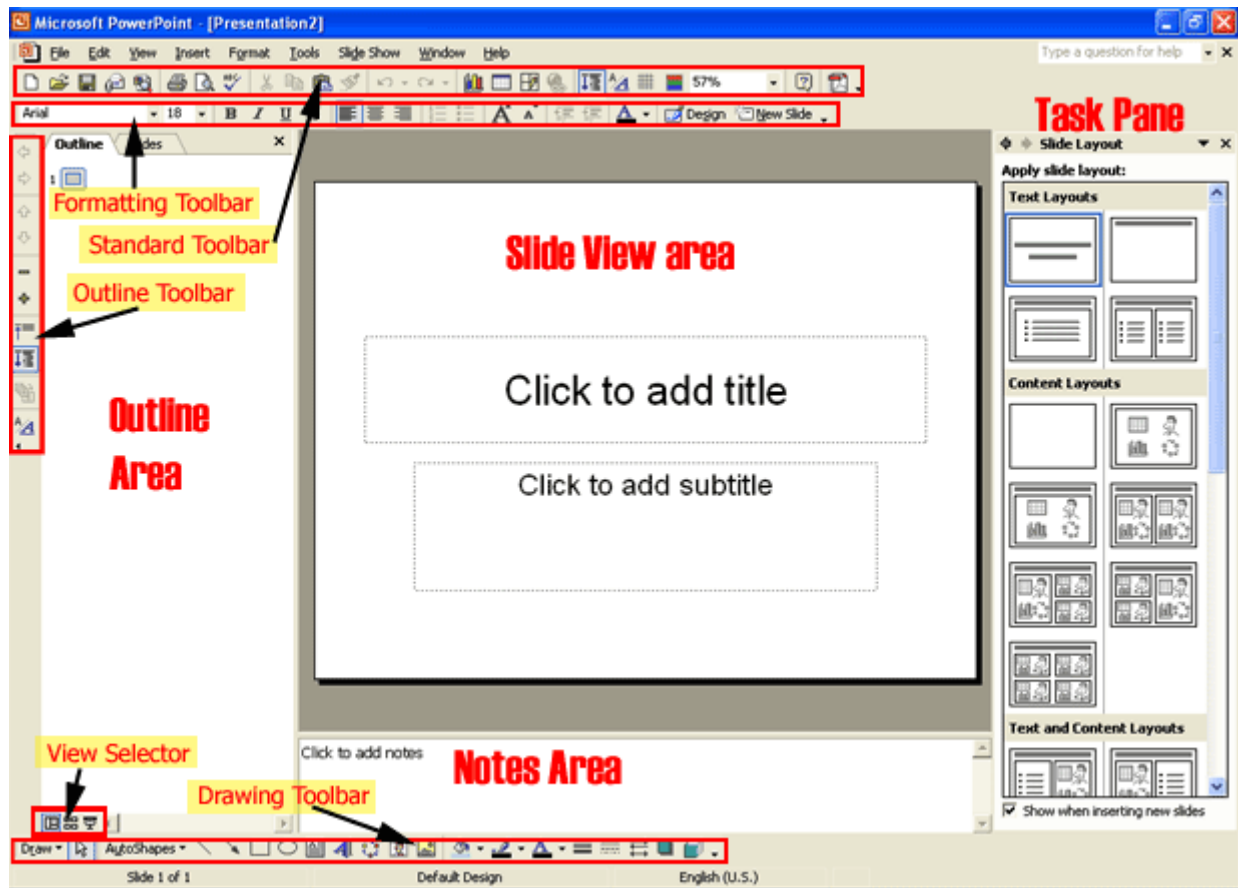


New Slide Layout Dialog Box

8.3. PowerPoint Window Properties:

The Microsoft PowerPoint window has many of the same attributes as other windows in the Microsoft Office Suite. The table below lists the different features and provides a brief description of each one.

Besides the usual window components, the PowerPoint window has several unique elements, identified in the figure below.

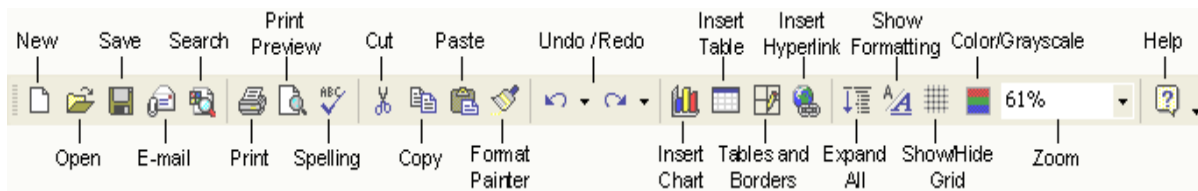


Menu Bar

The words listed atop the application window, immediately underneath the Title bar. You can access all application commands from the Menu bar

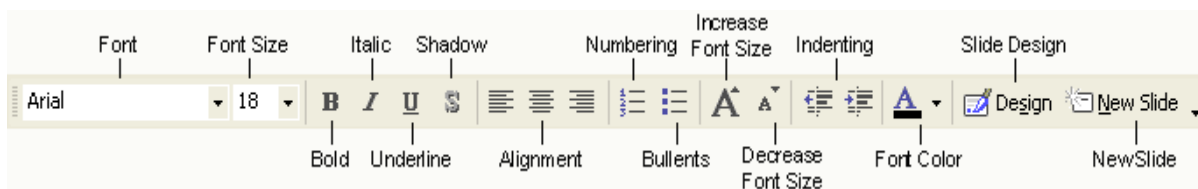
Standard Toolbar

The Standard toolbar, located beneath the menu bar, has buttons for commonly performed tasks like printing, saving, inserting clip art, and other operations. You can customize the toolbar or even display multiple toolbars at the same time.



Formatting toolbar

The Formatting toolbar, located beneath the Standard toolbar bar, or possible to the side of it, has buttons for various formatting operations like changing text size or style, changing alignment, formatting bullets, and animation



Drawing Toolbar

The Drawing toolbar on the bottom of the PowerPoint window contains drawing and text tools for creating graphics.



Slide Layout Task Pane

The area on the left side of the window that allow you to easily switch between the **Outline** and the **Slides** views.

Slide Area

The area in the middle, in which you actually work on your presentation slides

Task Pane

The area on the right side of the window that allows you to easily access various commands. You can display different Task Pane options by clicking on the dropdown arrow in the upper right-hand corner of the pane.

8.4. Save a PowerPoint Presentation:

Changes you make to a document are not saved to disk until you issue a Save command. Saving is quick and easy, and you should save often to minimize the loss of your work. PowerPoint has two save commands, Save and Save As, that work similarly. Both commands are on the File menu.

Save

When you save a new presentation for the first time, PowerPoint displays a dialog box similar to the Open dialog box. Select the disk in which to save the presentation and specify a name for the file. When you save an existing document that you have been editing, the newly saved version replaces the older version.

Save As

This command displays a dialog box where you can choose a document name and destination folder or disk. Use the Save As command whenever you want to save a copy of the current document with a different name or in a different folder or disk. The newly saved copy becomes the active document

A presentation is normally saved as name.ppt file type. However, PowerPoint Show with the extension of name.ppt is also a useful file type so that your file is able to run regardless of operating system.

8.5. Copy and Move data in a PowerPoint Presentation:

Use the Copy and Paste commands to copy selected text and graphics from one slide to another. If you want to move data instead, use the Cut and Paste commands. To do this:

- Select the text you want to cut or copy by highlighting it.
- Go to the Standard Toolbar to choose the Cut or Copy short cut icon.
- Move and click your mouse to the place where you want the text to go. Note the cursor is blinking.
- Go to the Standard Toolbar to choose the Paste short cut icon.

8.6. View Options:

Power point offers different views for entering, editing and previewing your information. They are:

- Slide view
- Outline view
- Slide Sorter view
- Notes Pages view
- Slide show view

Slide View

Slide view shows how our listed slides will work. We can move from one slide to another with **pg up** and **pg dn** keys. We can edit text and other slide elements in slide view.

Outline view

It will display all the text in the presentation. Here we can easily rearrange line item. We collapse items in this view so that we can see important headings or names of each slide.

Enter and Edit Text in Outline View:

PowerPoint's Outline Area lets you focus on the content of your presentation, by typing up the outline prior to laying the presentation out. After you type an outline, you can concentrate on how your presentation looks

The boundary of the Outline Area can be resized by dragging the vertical separation to the right. This will make it easier to enter your outline.

Entering Slide Titles and Bullets:

Typing in PowerPoint's Outline view is similar to using a word processor. Type your slide title or bullet point and press Return to start a new slide or bullet. To force a line break within a slide title or bullet, press Shift-Enter. This starts a new line for the same title or bullet.

Move Slides in the Outline Area:

It's easy to change the order of your slides and bullets in the Outline Area. Press the mouse button on the item you want to move and drag it to the desired location. You can also select the item you want to move and click on the Move Up or Move Down buttons.

Delete Slides or Bullets:

While working in the Outline Area, you can delete a slide by clicking on the slide icon next to the title and pressing the Delete key. Delete bullets and sub bullets similarly.

Slide Sorter view

It will show the slides in reduced size. We drop the sliders to move the slides so that we can change the order of slides. We can make the slides appear and disappear using the effects.

Notes Page View

It is to create and see notes to the presenter. It shows miniature slide image and provides a text area for presenters notes.

To enter notes

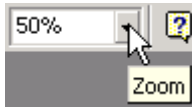
1. Click on the text area shown below this line. The box outline will change appearance.
2. Type and edit in the note box.

Slide show view

It will show the slides in reduced size. We drop the slides to move the slides so that we can change the order of slides. We can make the slides appear and disappear using the effects.

When you right click on the slide it will show some options. In that we can opt 'Pen' to draw the temporary lines. These lines will not be saved on the slides. To leave from the slide show press Esc.


Zoom



The **Zoom** adjuster lets you alter how large your slides appear on the screen relative to full-screen presentation mode. Click on the **Zoom** drop-down list (in the toolbar near the top of the screen), or click on the **View** menu, and select **Zoom...** Select "fit", a default percentage, or type in your own number, according to your needs.

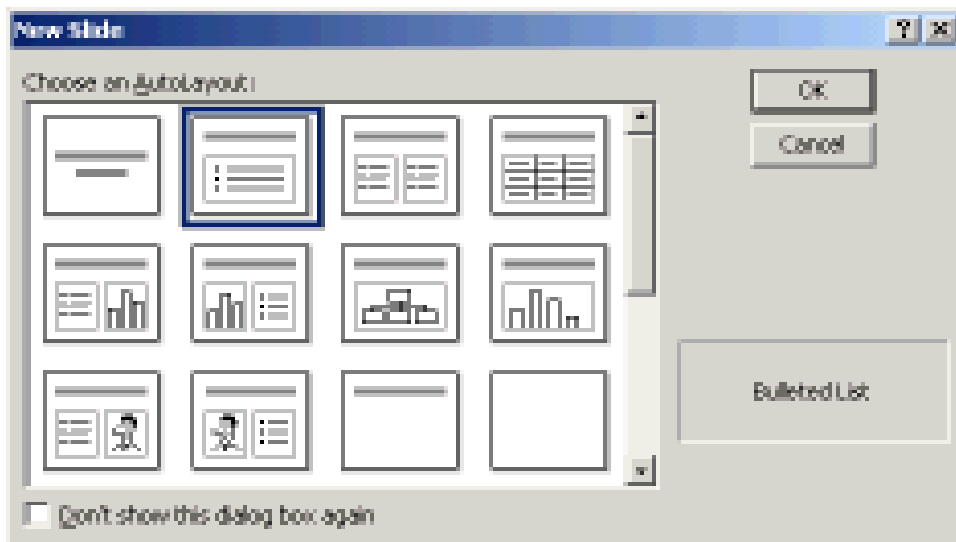
8.7. Add a Slide:

In the **Text Outline** area, you can add a slide *before* or *after* an existing slide.

- To add *before* a slide position the cursor at the beginning of text of the slide in front of which you would like a new slide, and press the **Enter** key.
- To add *after* a slide, position the cursor at the end of the text of the slide after which you would like a new slide, and press the **Enter** key (if you had indented levels of text on the previous slide, use the **Promote** (left arrow) button in the toolbar at the top of the screen to move the text back to the left to force a new slide. 

In **Normal**, **Outline**, **Slide**, or **Slide Sorter** view, select the slide *after* which you wish to insert a new slide.

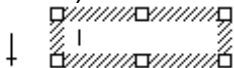
- Then click on the **Insert** menu, and select **New Slide**.
- When you add a new slide this way, the **New Slide** window appears, asking you to select a slide layout.
- Select whichever layout is appropriate for the content you will place on the new slide. The one selected in the example on the right is "Bulleted List". If the provided layouts don't meet your needs, select the bottom-right layout, "Blank", which will provide you with a new empty slide. However, no matter which layout you choose, you will be able to modify it later.



8.8. Add/Modify Text:

Here are a few ways to insert or modify text in PowerPoint:

- In **Normal** or **Slide** view, select the slide on which you wish to insert text. If there is already a text box on the slide you may click your mouse on it and start typing. Familiar word processing tools for modifying text (color, style, font, alignment, etc.) are provided.
- In **Normal** or **Slide** view, to add a new text box to a slide, click on the **Insert** menu, and select **Text Box**. Put your mouse over the slide layout - the cursor will change to a narrow down-arrow (as shown on the left below). Click, hold, drag, and release the mouse to form a rectangle on the slide. Then you can type into the text box created for you (as shown below).



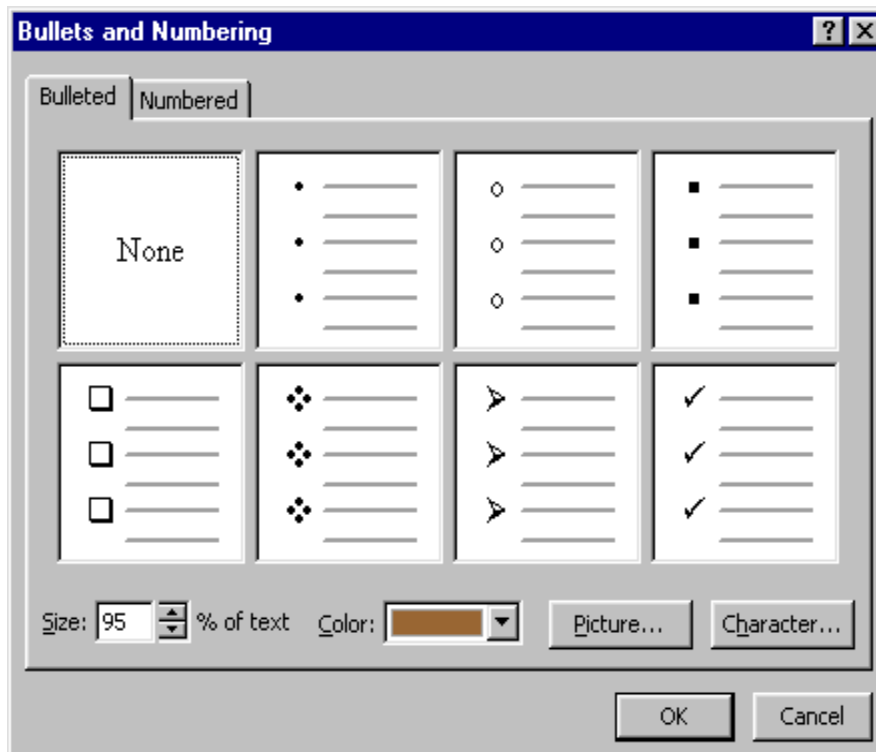
- To work directly in the **Text Outline** area, position the cursor at the point where you would like the new text to appear and just type it in. Use the **Promote / Demote** buttons in the toolbar at the top of the screen to create different levels of indentation. ← →

8.9. Bullet Styles:

Although you can set Bullet Styles in the Master, you also can change the bullet style on any slide or on any bulleted point. When you change the bullet style, every subsequent bullet on that slide, whether promoted or demoted, will have that style until you change back to the default or change to a different slide.

To change the bullet style, make sure the cursor is in the line of text in which you wish to change the bullet, then choose **FORMAT: Bullet**.

NOTE: Sometimes it is difficult to remember exactly what the default bullet style was, so pay close attention to the selected bullet before proceeding with any changes.



Format Bullet Dialog Box

You can choose from the selections provided by PowerPoint, choosing both the size (in percentage of text size) and the color.

Click Character to include any symbol font installed on your system. Keep in mind, however, that if you choose a font that is not installed on the computer from which you are giving your presentation, the presentation computer will choose the "closest" font to the one you chose and the result may not be one you like. Both Symbol and Wingdings font are standard on most computers and should display as you intended.

Adding / Removing Bullets



The Bullet Tool is a toggle. Click it once to place a bullet at the beginning of the current line of text (the one in which the cursor is flashing). Click it again to remove the bullet. This way you can add bullets to text you have typed without using the Bullet AutoLayout.

8.10. Speaker Notes:

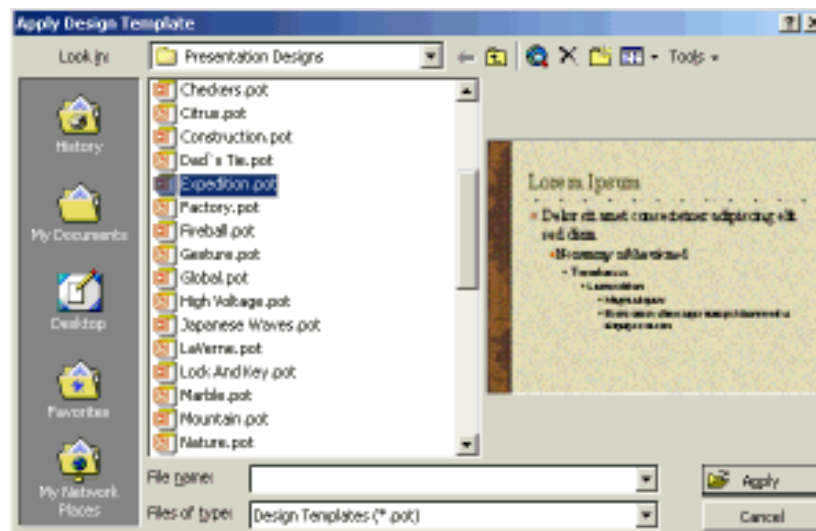
PowerPoint has a facility for adding speaker notes to a presentation. These can be notes that you create for yourself — for example, to help you remember key points during a presentation — or you can create them as additional information for your audience. After you have completed a presentation, you can print the notes for each slide, along with a smaller version of the slide. To add notes to a slide:

- Click on the **View** menu, and select **Notes Page** (you may want to adjust the **Zoom** percentage to get a larger view).

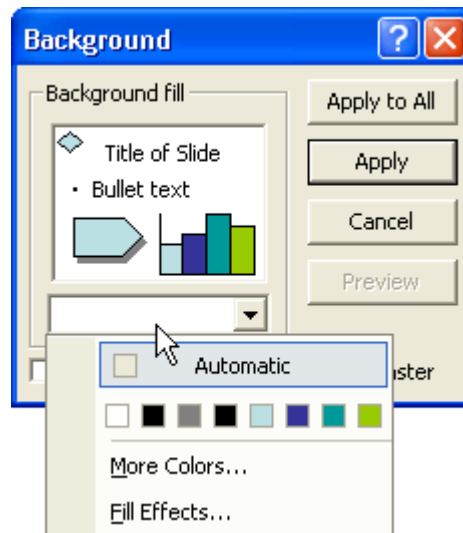


8.11. Background / Color Scheme:

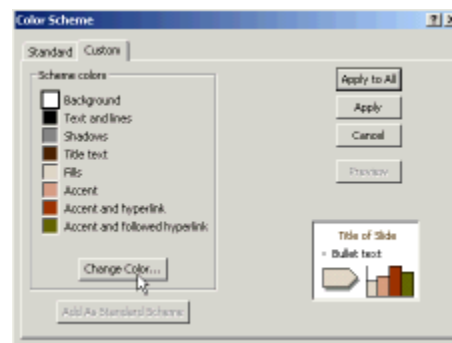
Design templates contain color schemes, slide and title masters with custom formatting, and styled fonts, all designed to create a particular look. Not only can you specify a design template when you begin a new presentation, you can choose a design template (for the whole presentation, not individual slides) any time while editing a presentation. To do this, click on the **Format** menu, and select **Apply Design Template**. A window will open asking you to choose a design. Select one and click **Apply**. The name of the template chosen will be displayed in the center of the status bar at the bottom of the PowerPoint screen. If the status bar is not visible, click on the **Tools** menu, select **Options**, click on the **View** tab, and ensure that **Status bar** is checked.



The *background color of slides* can be customized by clicking on the **Format** menu / **Background**. Click on the drop-down list to select either another color from the subset provided or click on **More Colors** and **Fill Effects** to see other choices. One of the **Fill Effects** available is **Picture** (import an external image file to use as a slide background). When you are done, click on **Apply** (if you wish to alter just the current slide) or **Apply to All** (to alter all slides at once).



The *color scheme for slides* can be customized by clicking on **Format / Slide Color Scheme**. Either chooses a preset scheme from the **Standard** tab or click the **Custom** tab, click on the desired component (e.g. Background), and click **Change Color** (a color selection window will appear). When you are done making color choices, click on **Apply** (if you wish to alter just the current slide) or **Apply to All** (to alter all slides at once).



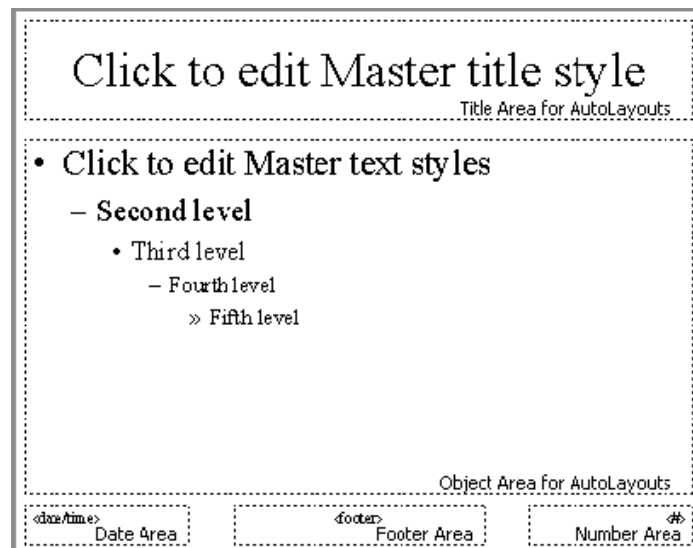
8.12. Slide Master

The Slide Master in a template is a slide that controls the formatting, text, and objects that appear on every slide in your presentation. For example, if you want a small picture of the world to appear on every slide, place that picture on the Slide Master. To display the Slide Master, choose

Master/Slide Master from the View menu. You can then edit this slide. Changes you make to the Slide Master also appear on each slide in your presentation.

It is easy to edit the text of a slide in the Outline Area. However, to modify a slide's appearance, switch to Slide View. If you want your changes to apply to every slide in the presentation, choose Master/Slide Master from the View menu.

In Slide View just click on an object to select it. Then use the Format menu to apply the change you want. For example, to change the font or color of the slide title, select the slide title and choose Font from the Format menu. The Format menu also has commands for centering or left aligning text (Alignment), and changing colors (Colors and Lines). You can even change the format of slide bullets using the Bullets command on the Format menu. PowerPoint also has a Formatting toolbar to simplify basic formatting tasks.

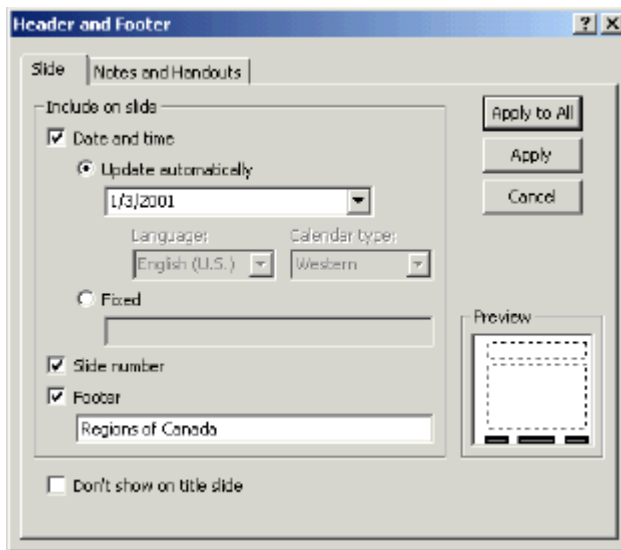


Sample Slide Master

8.13. Headers / Footers:

- On the **View** menu, click **Header and Footer**.
- On the **Slide** tab, you set options, the effect of the options selected will appear at the bottom of slides:
 - **Date and time** can be set to **Update automatically** (to reflect the last date the slides were modified) or **Fixed** (type a certain date such as when the presentation will be given).
 - Selecting the **Slide number** option will automatically insert a sequence number.
 - Selecting **Footer** (and typing text in the box below it) inserts fixed footer text.

- Selecting **Don't show on title slide** will omit the footer from the title slide (but it will appear on the other slides).
- Normally you click on the **Apply to All** button to make these changes throughout the presentation, but of course you can click **Apply** to affect just the current slide.
- Similar options are available on the **Notes and Handouts** tab, plus the option of adding a page header to your printouts.



8.14. Summary:

At the end of this lesson you have learnt the important features of Ms-PowerPoint including creating a presentation, opening an existing presentation and working with different kinds of slide layouts. We identified different toolbars available in PowerPoint.

You are in a position to format the text, applying background color to the presentation, working with master slide, applying speaker notes to slide, applying header and footer to the document.

8.14. Technical Terms:

Slide: Brief, concise, descriptive phrases that will help you remember what you want to present and to serve as a reminder for your audience.

PowerPoint Presentation: PowerPoint presentation consists of slides that can contain text, graphics, charts, and other data types.

Slide Master: The Slide Master in a template is a slide that controls the formatting, text, and objects that appear on every slide in your presentation.

Speaker Notes: Speaker notes can be notes that you create for yourself, key points during a presentation, additional information for your audience.

8.15. Model Questions:

1. What are the features of PowerPoint?
2. What is a Slide? Explain the procedure to create a Slide?
3. What is Slide Master? Write the procedure to create Slide Master?
4. Explain the procedure to create Header and Footer in PowerPoint?
5. Explain the views available in PowerPoint?

8.16. References:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, 'The ABC'S Of Microsoft Office.' BPB Publications, New Delhi

MS OFFICE 2000 Complete, BPB Publications, New Delhi

C.V.P.R.PRASAD, M.C.A.,

Lecturer, Dept.Of Computer Science,

JKC College, GUNTUR.

Lesson – 9

MICROSOFT POWERPOINT

9.0 Objective:

This lesson introduces the Drawing Objects, Images and ClipArt, Audio and Video, Slide Transition, Rehearse Timings and Charts.

Structure of the Lesson:

- 9.1. Working with Text
- 9.2. Drawing Objects
- 9.3. Images and Clip Art
- 9.4. Audio and Video
- 9.5. Creating Hyperlinks
- 9.6. Animating the presentations
- 9.7. Slide Miniature
- 9.8. Slide Transition
- 9.9. Slide Show Animation
- 9.10. Rehearse Timings
- 9.11. Slide Setup and Printing
- 9.12. Charts
- 9.13. Summary
- 9.14. Technical Terms
- 9.15. Model Questions
- 9.16. References

9.1. Working with Text:

Any time you wish to make formatting changes to text, you must highlight the text to be changed. Making a selection from the Text Toolbar affects the highlighted text only. In addition to the standard text formatting tools of font choice and size, bold, italics, and underline, you also can align text left, center, or right. Other special tools are described below.

Shadowed Text


The Shadow Text Tool places a slight shadow behind the text. Although the effect is not prominent, this is an excellent tool to use to give more definition to title text. You should NOT shadow all body text since this makes it very "muddy" to read.

Hollingsworth
shadowed

Hollingsworth
not shadowed

Sample Shadowed Text


To shadow text,

1. Type the text to be shadowed.
2. Highlight the text.
3.  Click the Shadow Tool.

Increase / Decrease Paragraph Spacing

 The Increase and Decrease Paragraph Spacing Tools expand and contract paragraph spacing, respectively.

Format Painter

 The Format Painter Tool picks up any formatting applied to existing text and applies that same formatting to any text it is "painted" over. The two pieces of text need not be on the same slide.

To use the Format Painter,

1. Highlight the text containing the formatting you desire to "pick up".
2. Click the Format Painter Tool.
3. Press and drag over the text as you would to highlight it. The formatting will be applied.

After the formatting is applied the Format Painter will de-select.

To apply formatting to multiple selections of text without needing to click the Format Painter between each application, double click the Format Painter. The Painter will remain the selected tool until you click it to de-select, allowing you to press and drag over multiple sections of text, applying the same format to each.

Increase/Decrease Font Size



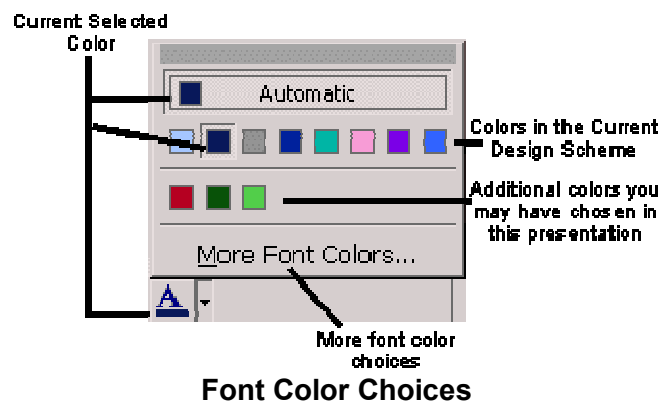
You can increase or decrease the font size of highlighted text by clicking on the Increase Font Size or Decrease Font Size Tool, respectively. Size increments and decrements by the standard font sizes.

Text Colors



You choose text colors with the Font Color Tool located in the Drawing Toolbar. Highlighted text will change to the color displayed in the band beneath the A.

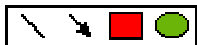
To make additional color choices or to see additional non-scheme colors you have used in this presentation, click the drop-down arrow beside the A.



Word Wrap

Whether text is in a Placeholder or a textbox, you can narrow or widen the width of the box by dragging the box border. This affects where the text wraps. Make sure the cursor looks like a two-headed arrow to affect the Placeholder width; a four-headed arrow will move the placeholder.

9.2. Drawing Objects:

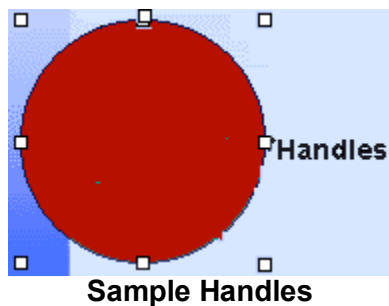


To draw a line, an arrow, a square, or a circle, select the appropriate tool from the Drawing Toolbar. When you move the cursor over the slide, it will be a crosshair. Press and drag the cursor to insert the object. The larger an area you drag, the larger the object will become.

TIP: To draw a straight line or arrow, or a perfect square or circle, hold SHIFT as you press and drag.

Selecting

To alter any PowerPoint object, you first must select the object. When you click the object, you will see Handles around the perimeter of the object. This lets you know exactly which object you will be affecting.



Deleting

To delete an object, click to select it, then press DELETE.

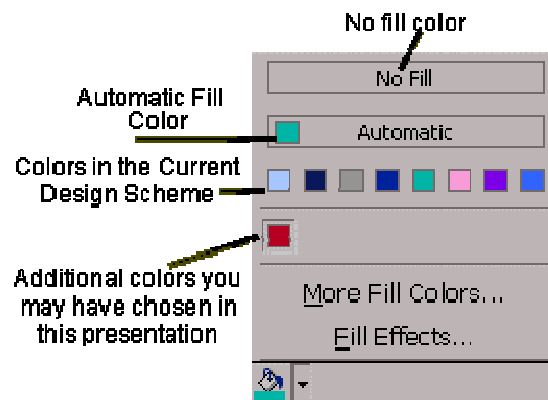
Fill Colors

When you draw solid objects (squares and circles), they fill with the automatic color. The underline under the Paintbucket is the current fill color.



To fill an object with the current fill color, select the object, then click the Paintbucket Tool.

To change the current fill color, drop down the arrow beside the Paintbucket and make a new selection. There are also additional options under Fill Effects, including choosing no fill color at all.

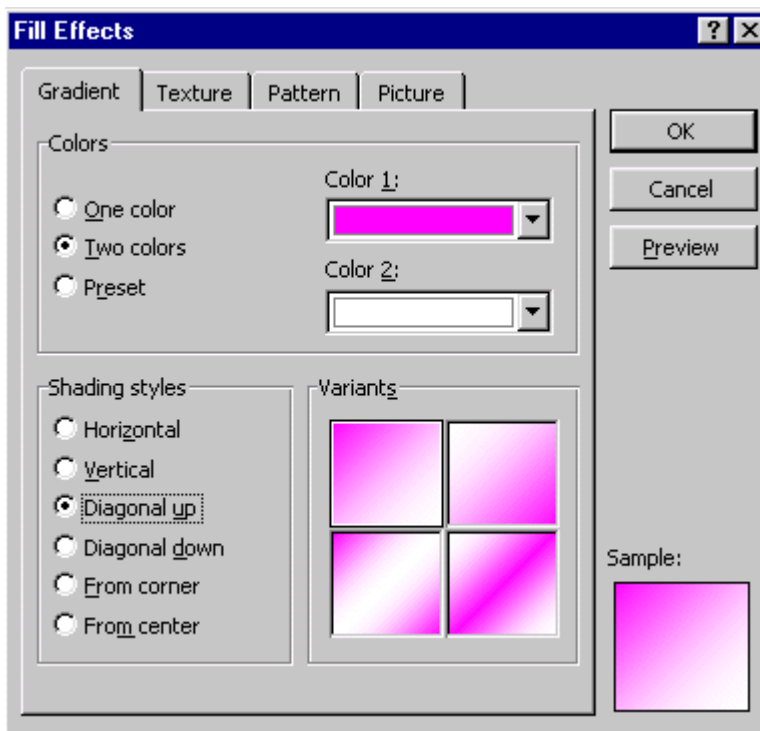


Alternate Choices for Fill Colors

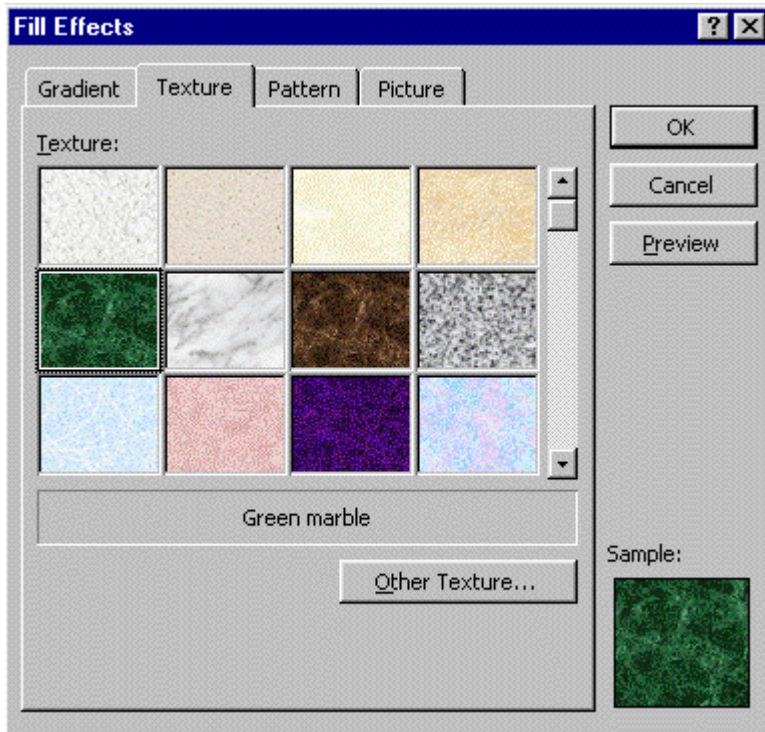
Fill Effects

Fill effects include Gradients where you can select one or two colors of your choice, as well as the direction and variants of the gradient. Textures are image files similar to web page wallpapers; you can use PowerPoint's selections or choose an image file from your workstation. Patterns allow you to choose two colors and the pattern design with which you want to fill the object. Picture allows you to choose an image file from your workstation; the photograph will center itself within the selected object, hiding any parts of the image, which do not automatically show within the object's shape.

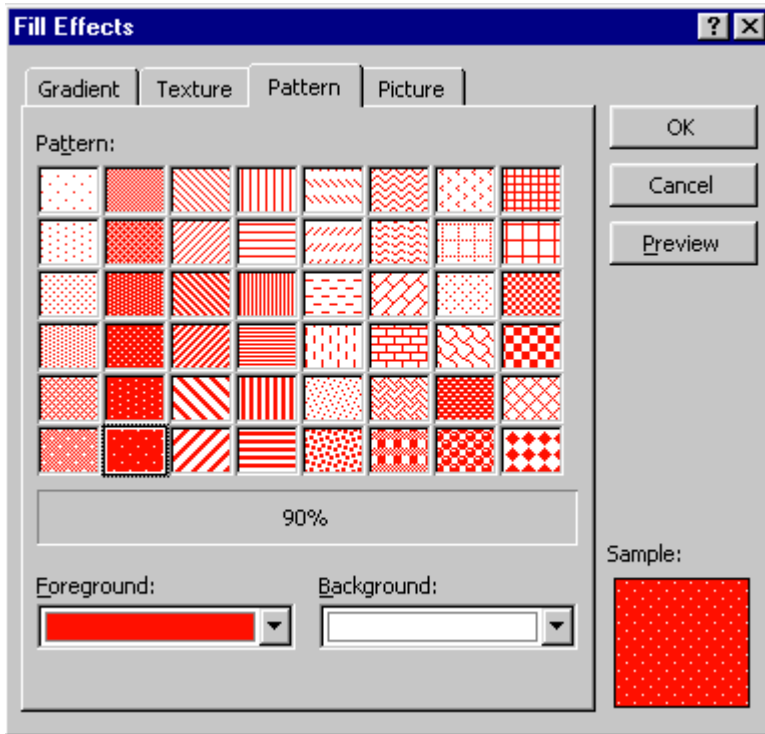
Gradients



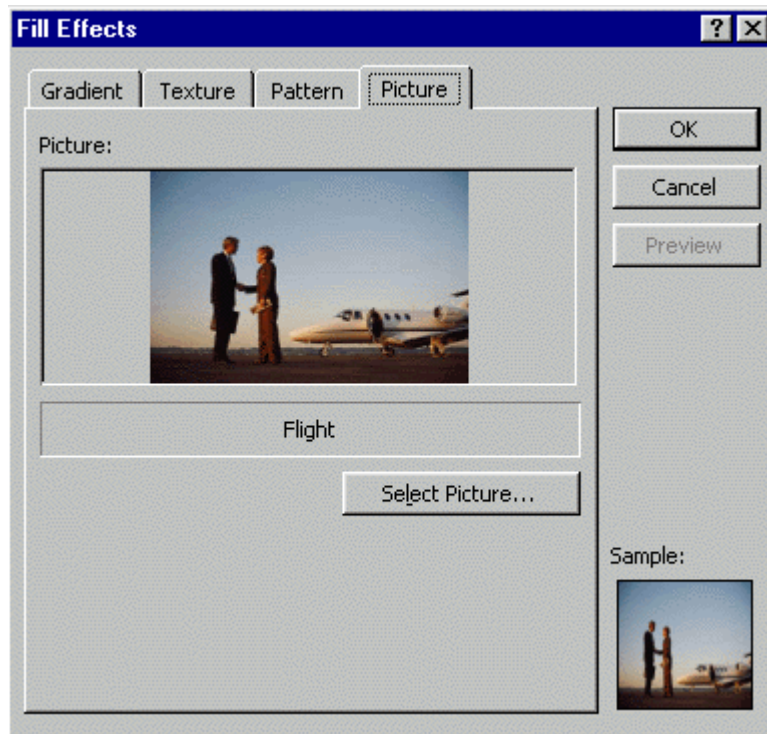
Textures



Patterns



Photos



Line Effects



Each drawing object has an outline the color of the automatic color. The underline under the Line Color Tool is the current line color. To outline an object with the current line color, select the object, then click the Line Color Tool.

To change the line color, drop down the arrow and make a new selection. Other options with the Line Color Tool include color and patterns similar to the Fill Tool.



To change line style, dash style, and arrow style, click on these tools, respectively, after drawing the object and while it is still selected.

Object Effects

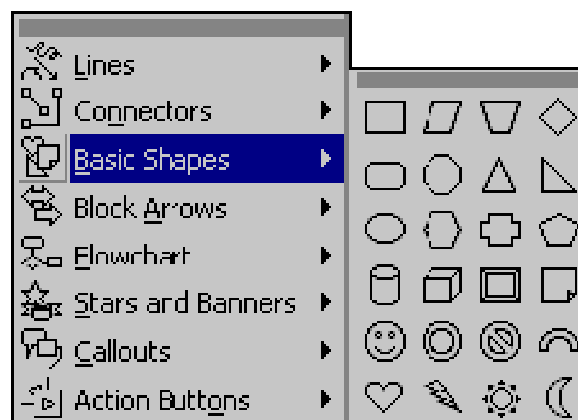


To place a shadow or 3D effect on an object, select the object, then click the tool selection.

AutoShapes and WordArt

If you're not particularly artistic, AutoShapes and WordArt can come to the rescue. To use AutoShapes,

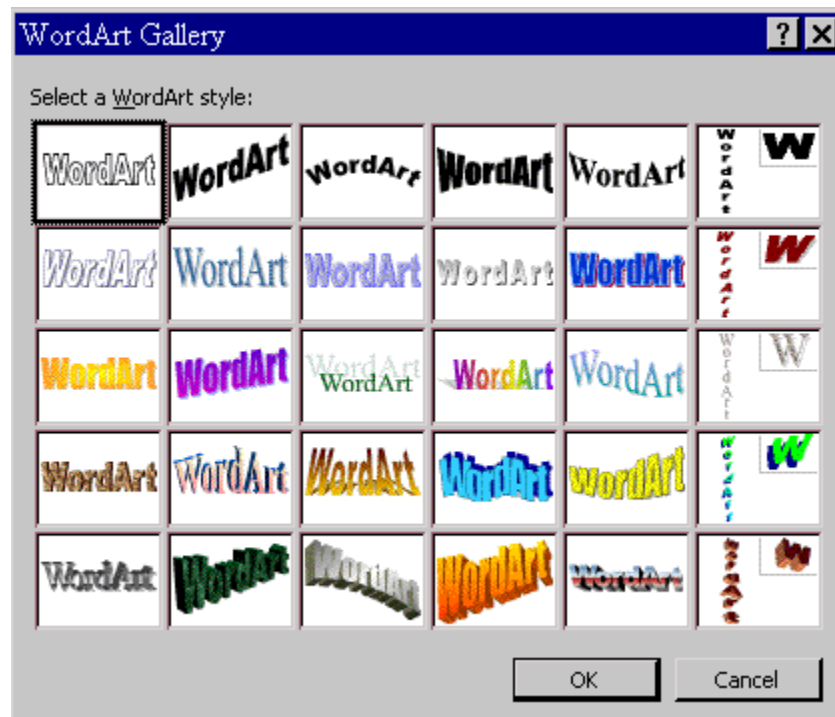
1. From the Drawing Toolbar, click the drop-down arrow beside AutoShapes, and then select the shape type.
2. Click the shape you want. The cursor will change to a cross-hair as you move the mouse across the slide.
3. Click on the slide where you want to place the shape.
4. Move the object, change its size or color, as desired.



Autoshapes Tool

 To use WordArt,

1. Click the WordArt Tool.
2. From the WordArt Gallery, click the design you want, then click OK.

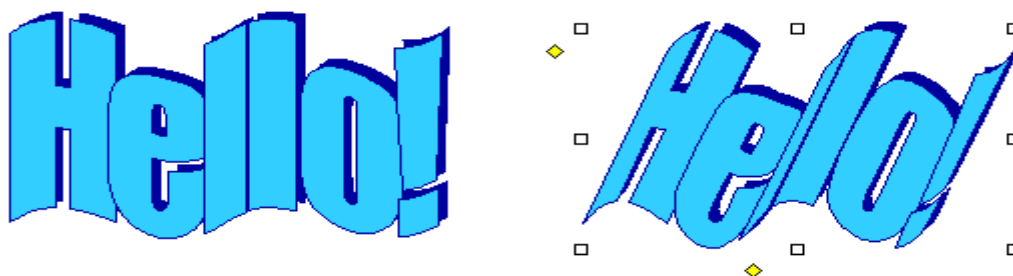


WordArt Gallery

3. In the Edit WordArt Text box, type your text. You can also change the font face and type size, or add bold or italics. Then click OK.

The art will be placed in the center of the slide and is always smaller than you would like—press and drag the handles to enlarge the object to the desired size.

Many of the WordArts also have one or more small yellow diamond-shaped handles. You can press and drag any of these to distort the shape of the object:



Samples of Word Art

To change the colors in WordArt, make sure the object is selected, then choose FORMAT: WordArt and make your selections.

Text Tool



To insert text on the slide other than in a Placeholder

1. Click the Text Tool.
2. Press and drag the cursor over the slide to create a Text Box.
3. Type in the box.

NOTE: Any text you type in a placeholder will show in Outline View; text inserted via the Text Tool is treated as an object and will not display in the outline.

Moving

To move an object on the slide, select it, then move the cursor over the object until you see the four-headed arrow along with the cursor. Press and drag the object to the location you need it.

NOTE: Be sure it is a four-headed arrow rather than a two-headed arrow. Two-headed arrows re-size objects.

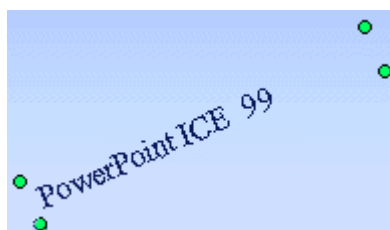
To move a text box, select it, then move the cursor over the frame until you see the four-headed arrow along with the cursor. Press and drag the text box to the location you need it. If you move the cursor within the box, it will change to an insertion point so that you can type or edit text.

Rotating



To rotate any object, including a text box,

1. Click the Rotate Tool.
2. Click the object to be rotated.
3. You will see small green disks in place of the normal handles on the object.



Sample Rotated Text

4. As you move the cursor over the disk, it will change to a looped circle. Position the loop over one of the green disks and press and drag to the desired rotation.
5. Click the Rotate Tool to deselect it.

Grouping and Ungrouping

Often after you have created multiple objects and layered them in the sequence you need, you want them to "stay put". By grouping multiple objects, you can move or resize them as a single entity rather than manipulating their individual components.

To group multiple objects:

1. Size each individual object as necessary.
2. Layer objects as necessary.
3. Each object must be selected prior to grouping. To select multiple objects, hold SHIFT then click each object. Each object clicked becomes selected and the handles of each are displayed. Release SHIFT after you have selected each of the objects to be grouped.
4. From the Drawing Toolbar, choose DRAW : Group. You will see that the objects' individual handle-sets will be combined, leaving a single handle-set for the grouped object.

Sometimes, if your object is very complex, you will find it beneficial to group your object components incrementally. When you finish a subset of your final object, group the components you have finished. Then you can continue building additional components, knowing you won't accidentally move or re-size the portion you have completed.

Text that you include in a grouped object will not change point size if you re-size the grouped object, however, word-wrapping may adjust to conform to the re-sizing.

To manipulate individual components of grouped objects, you must ungroup them. To ungroup an object, select it, then from the Drawing Toolbar, choose DRAW : Ungroup. After editing, you can choose Re-group.

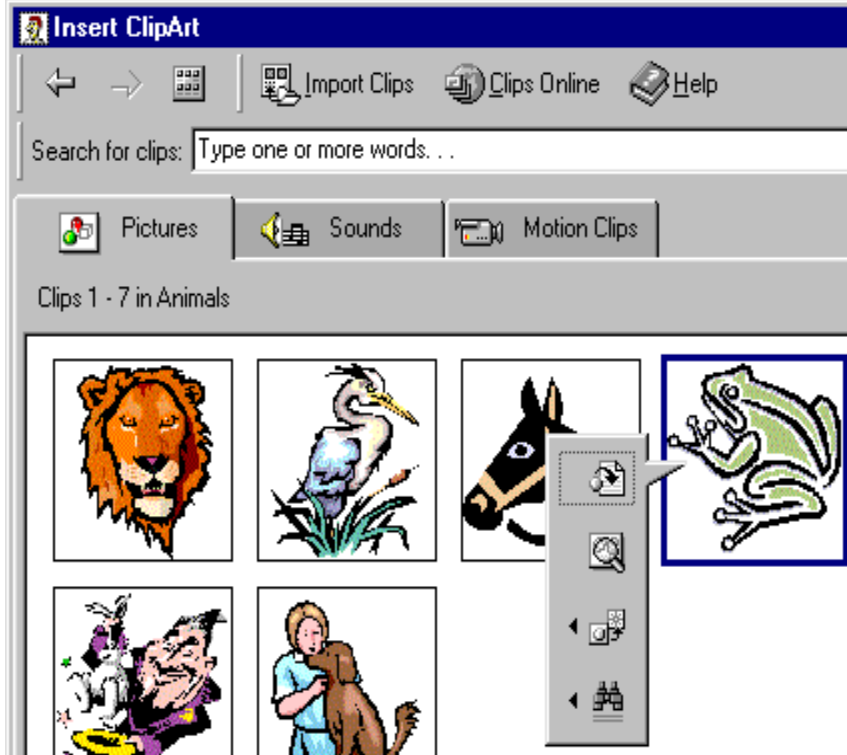
9.3. Images and Clip Art:

PowerPoint comes with a library of "clip art" that includes traditional clip art, as well as videos, photographs, and sounds. You can also insert graphics from other sources into a presentation.



To access PowerPoint's ClipArt Gallery, click the ClipArt Gallery Tool (located on the Drawing Toolbar).

To use a Gallery ClipArt image, in the Insert ClipArt dialog box, select the category pertaining to the type of image you are looking for, then select the image you want. A pop-up box will allow you to: insert the image, preview the clip, add the clip to a different category, and/or find similar clips.



Insert ClipArt Dialog Box

The greatest benefit of the ClipArt Gallery is that you can add multiple selections to your presentation without leaving the Gallery.


ClipArt images will be inserted in the center of the slide. You can move, enlarge, or reduce them.

To select multiple objects using the mouse,

1. Position your mouse cursor beyond the corner of the object to be re-grouped. Make sure you start far enough beyond the objects to catch all of the components.
2. Press and drag diagonally across the objects to select all the components. Make sure that you do not accidentally include an object in the selection that you do not want to be selected. (You may have to try this step several times, it is a bit tricky to master.)
3. After all the components are selected, from the Drawing Toolbar, choose DRAW : Group (or Regroup).

9.4. Audio and Video:

Your presentation can link to external audio and video files.

- Select the slide on which you wish to add a link to an audio or video file.
- On the **Insert** menu, point to **Movies and Sounds**.
- To insert a sound from a file, click **Sound from File**. To insert a video from a file, click **Movie from File**.
- A standard file directory window appears.
- Locate the folder that contains the audio/video file you want, and then double-click the filename.
Note: In PowerPoint, by default your movie or sound will play manually. If you want it to play automatically you will have to set **Custom Animation** options for the slide.
- For audio files, a sound icon  appears on the slide.
- For movies, an image consisting of the first frame is displayed as a placeholder.
- To preview the audio/video in Normal view, double-click its icon.

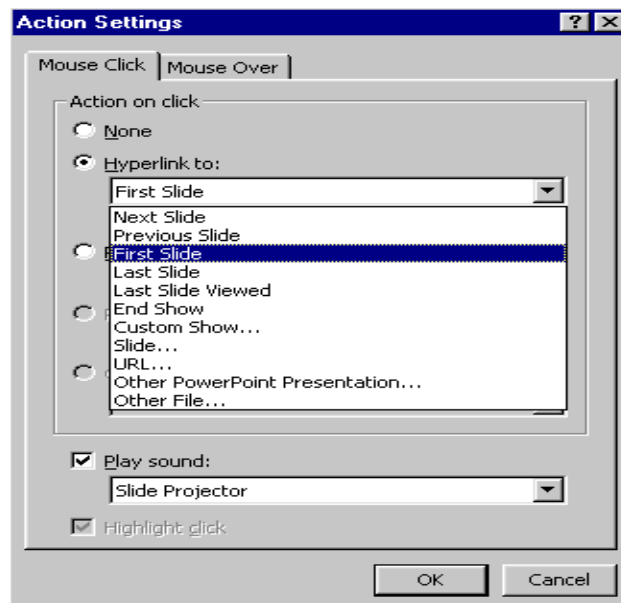
Note: Alternatively, you may have to use the **Insert / Object** command since PowerPoint is not able to play all audio/video formats inline. **Insert / Object** will open the media element in a new window rather than making it appear embedded on the slide, and it will be played by an external program.

9.5. Creating Hyperlinks:

Hyperlinks cause the selection of an object to result in a move to a new location or the performance of an action. You can create your own object or choose an AutoShape Action button.

To link an object with an action, select the object, then choose SLIDE SHOW: Action Settings.

In the Action Settings dialog box, choose either Mouse Click or Mouse Over. Then select the button "Hyperlink to" then select the destination or action. You can also choose to play a built-in PowerPoint sound during this action and/or to highlight the object as you click it.

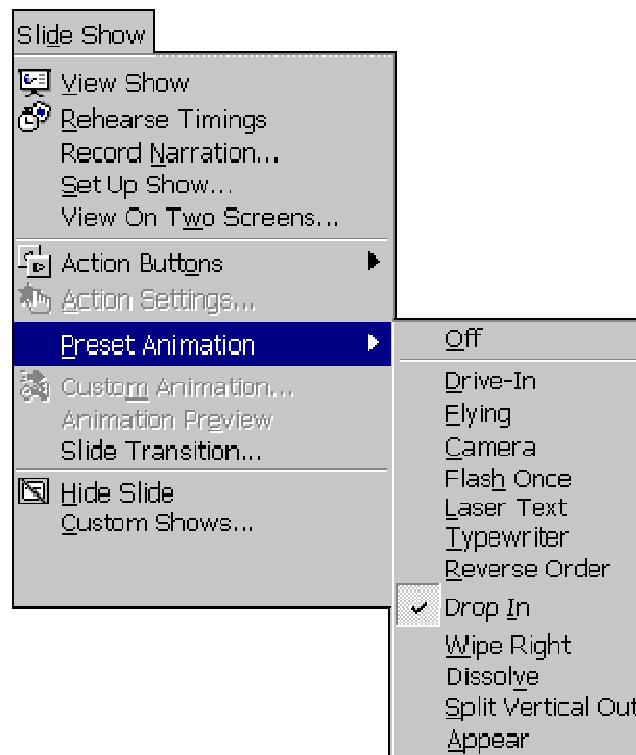


Hyperlink Action Dialog Box

9.6. Animating the Presentation:

To animate a single object on a PowerPoint slide

1. Select the object.
2. Choose SLIDE SHOW: Preset Animation.
3. Select the animation effect you desire for the selected object.



Preset Animation Dialog Box

Animated Layers or Slides

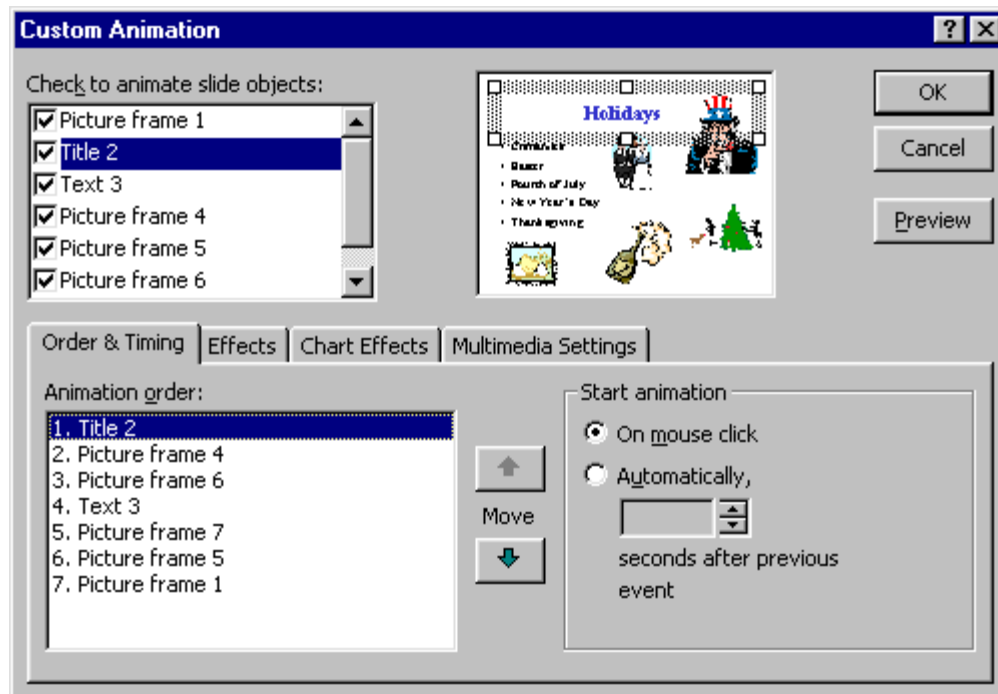
You may find it useful to bring a layered object into a presentation one layer at a time, building it on-screen. To animate a layered object or to animate each object on a slide

1. Choose SLIDE SHOW: Custom Animation. (You must be in Normal, Outline, or Slide View.)
2. In the Custom Animation dialog box, each object is identified in the Check to animate slide objects list.

If you don't remember what a particular object is, click the object's name in the list; that object appears selected in the preview window.

3. Click in the object's checkbox to animate that object. It will be added to the Animation order list.
4. To change the order that the animated objects appear, select the object in the Animation order list, then click the up or down arrow to move the object through the list.

5. Choose whether the object appears only on a mouse click or after a specified number of seconds.



Layer Animation Dialog Box

To choose other effects such as sound effects and how the object appears on the slide during the on-screen presentation, click on the Effects tab.

To animate text,

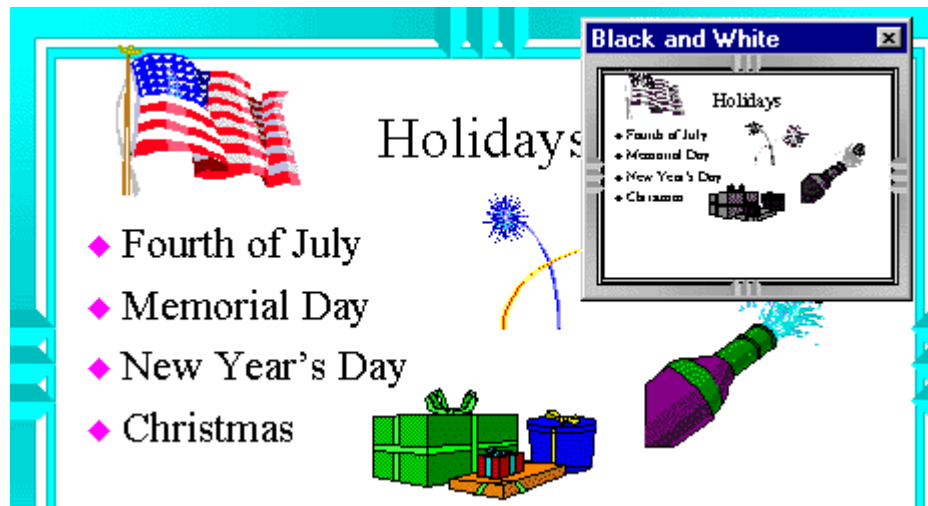
1. Click on the Effects tab.
2. In the Introduce Text section, from the drop-down box, choose whether the text is introduced all at once, by word, or by letter.

To animate charts click the Chart Effects tab. You can choose to introduce the data by series, categories, or elements.

NOTE: Be careful of using too many effects in any single presentation because they soon become a distraction rather than an enhancement of the presentation.

9.7. Slide Miniature:

When you have zoomed the view of your slide so that all of the slide cannot be seen on the screen at one time, PowerPoint will create a black and white Slide Miniature in the corner. This allows you to see the overall effects of the changes you are making to the slide.



Sample Slide Miniature

You may also choose to work in black and white mode by choosing VIEW: Black and White. To see a small color view of the current slide when you are working in black and white, choose VIEW: Slide Miniature.

9.8. Slide Transitions:

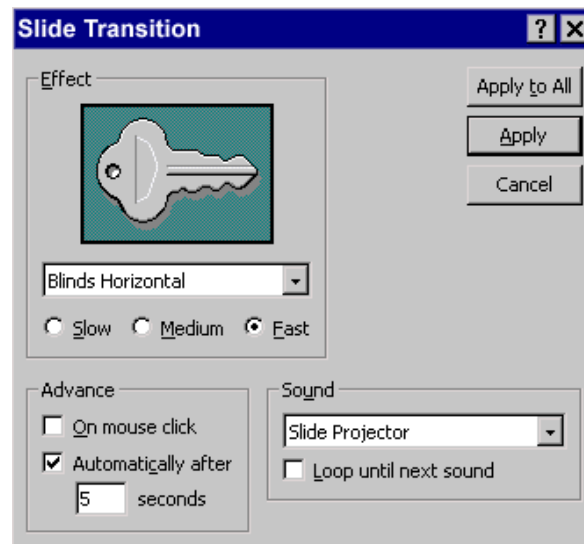
Transitions determine the effects applied when you move from one slide to another during an on-screen presentation. You must be in Slide Sorter View to access the Transition Toolbar.

To choose only a transition effect, click the dropdown box beside the words "No Transition" and select a transition. To have other transition choices available to you, click the Slide Transition Tool.



Sample Transition Tool and Drop-down Box

When you click the Slide Transition Tool, the Slide Transition dialog box allows you to choose not only a transition effect from a drop-down box, but also offers other choices that affect your on-screen presentation.



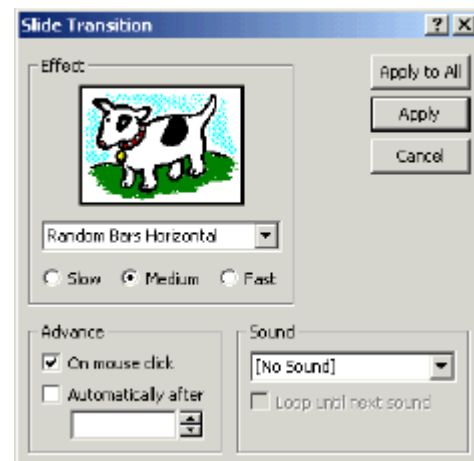
Sample Transition Dialog Box

TIP: Do not use a different transition on each slide because your audience attention should remain focused on you and the content, not the effects. If you want to use multiple transitions, a good rule of thumb is to apply a unique transition to each slide in a specific module of the presentation.

Generally, you should leave the Effect Speed set to Fast. This brings in the next content slide quickly and keeps your audience's attention focused on the presentation content.

PowerPoint contains a few built-in sounds. To play a sound as the slide is transitioning, make a sound selection from the Sound drop-down box. Unless you have a well thought out reason for doing so, do not choose to have the sound loop until the next sound; it can become annoying very fast while you are giving a presentation, as well as making it difficult for you to speak over. Also, many presentation locations do not have adequate speakers attached to the computer to allow all of your audience to hear the sound.

- In **Slide** or **Slide Sorter** view, select the slide or slides on which you want to add a transition.
- On the **Slide Show** menu, click **Slide Transition**.
- From the **Effect** drop-down list, click the transition you want.
- Select any other options you want (speed Slow/Medium/Fast, Advance on mouse Click or Automatically, Sound or No Sound)

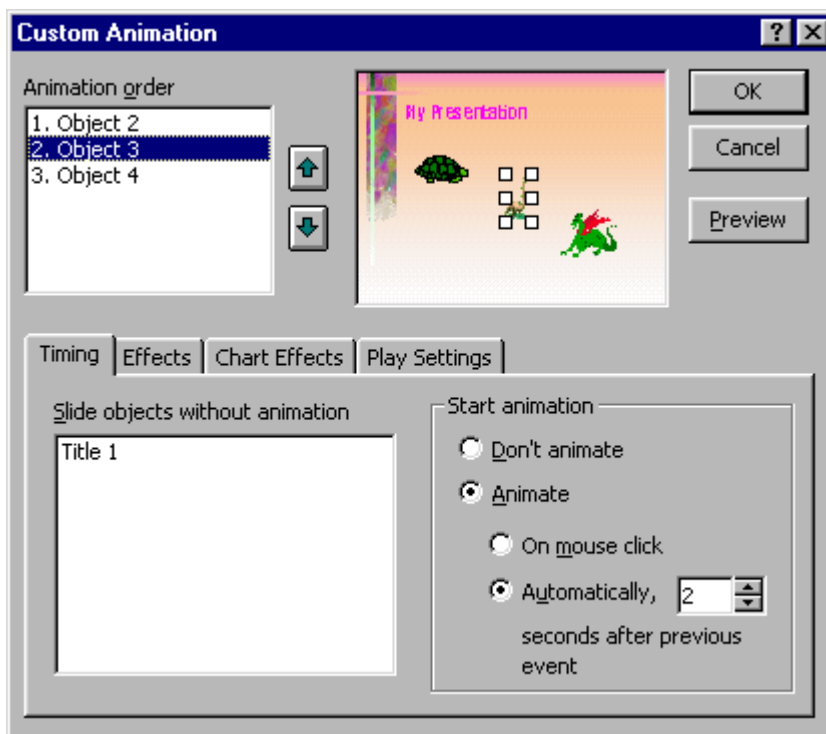


- To apply the transition to the selected slide, click **Apply**. To apply the transition to all the slides, click **Apply to All**.
- Repeat the process for each slide you want to add a transition to.
- To view the transitions, on the **Slide Show** menu, click **Animation Preview**.

9.9. Slide Show Animation:

You can control how and when text, graphics, and other objects appear on your PowerPoint slides. For example, you can make graphics or text items appear one at a time and add visual or sound effects. Animation can aid in focusing attention during a presentation and make your slide show more interesting. To set slide show animation options, display the slide that has the objects you want to animate, then on the **Slide Show** menu, click **Custom Animation**.

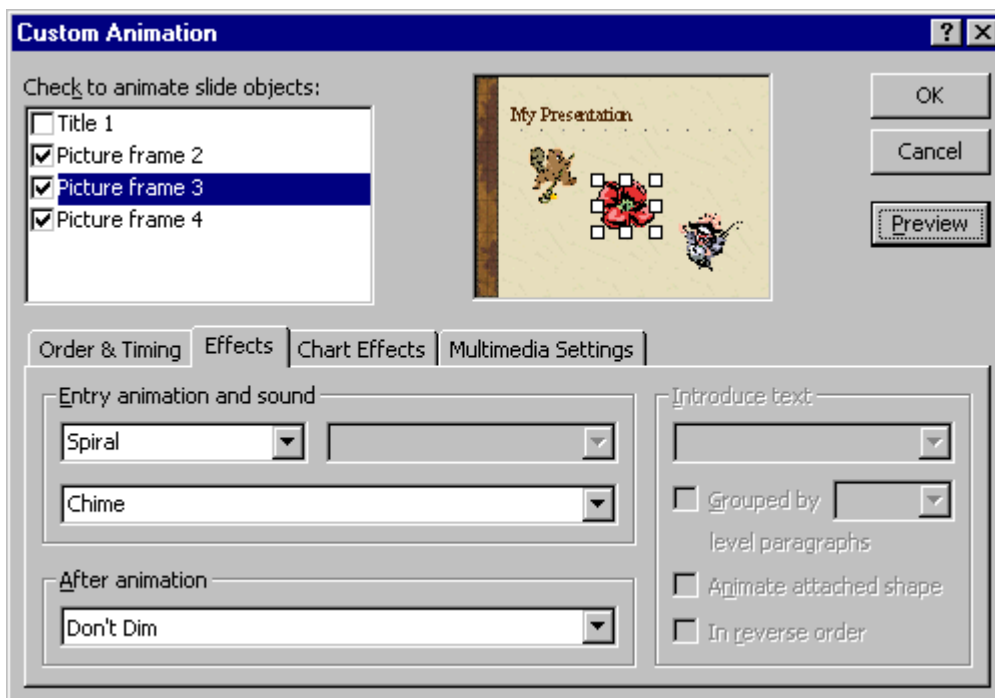
Timing Tab



- In PowerPoint, ensure that you are on the **Timing** tab.
- Under **Slide objects without animation**, double-click the objects you want to animate. (they should be moved up to the **Animation Order** box).
- Use the up and down arrows to the right of **Animation Order** to change the sequence in which objects will appear (move objects up or down in the list).

- Under **Start Animation** set the timing (when objects will appear). Select each object and then do one of the following:
 - To start the animation by clicking the object, select the **On mouse click** option.
 - To start the animation automatically, select the **Automatically** option, and then enter the number of seconds you want to have elapse since the previous animation.

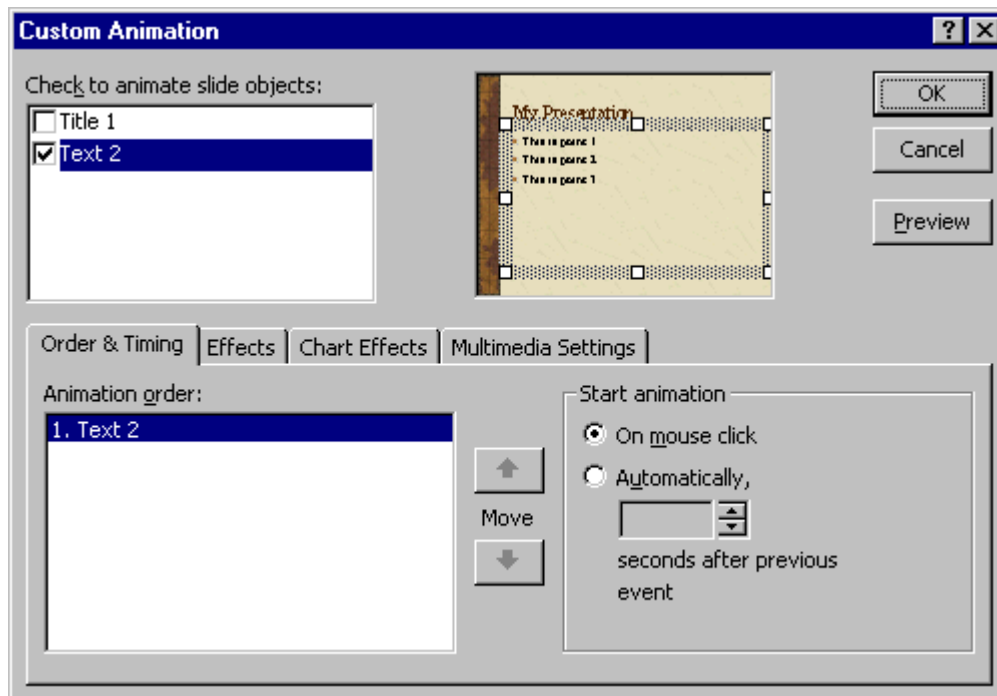
Effects Tab



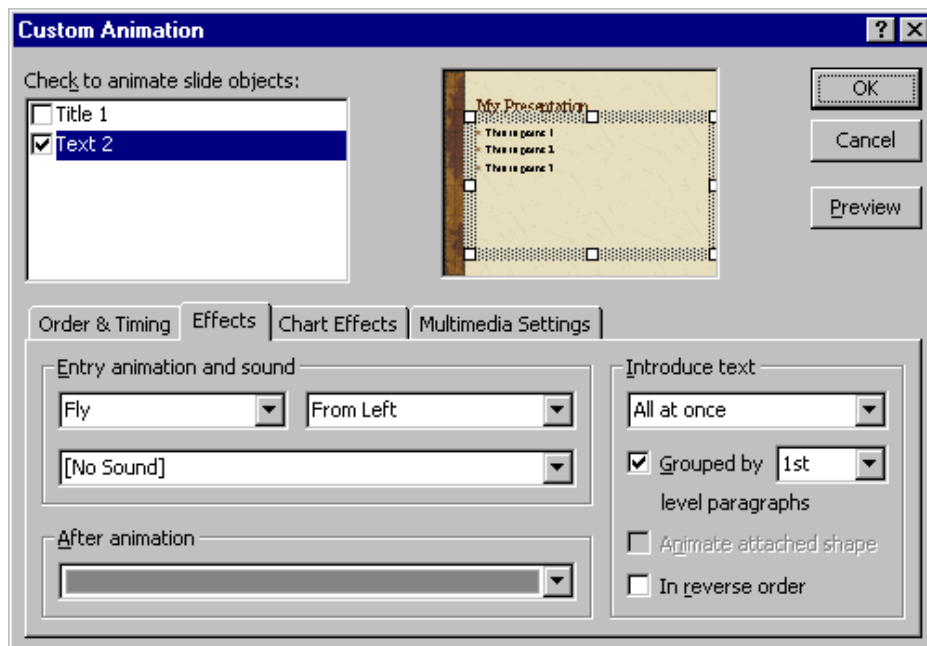
- Click the **Effects** tab to set options for visual or sound effects.
- Under **Entry animation and sound** select the options you want.
- To preview the animation effects, click the **Preview** button.
- In the example above, the object will enter the screen with a **Spiral animation** and the **Chime sound** will play.

Animating a text box containing bulleted text

- On the **Order & Timing** tab, the text box is selected with a check mark, and the **On mouse click** option is chosen (to let the presenter control the slide show with mouse clicks).



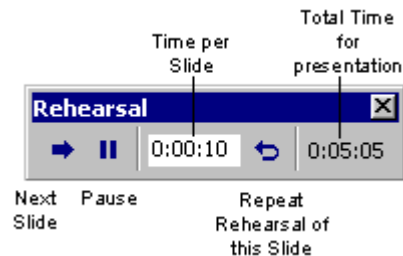
- On the **Effects** tab, the options set below will make each line of text enter the screen with a **Fly From Left** animation, **No sound** will play, and when the next line of text appears the previous line will change to a lighter color (a **dimming** effect). To preview the animation effects, click the **Preview** button.



The series of images below show what the screen looks like after each mouse click when the slide with the bulleted text (set with options as above) is displayed during a **Slide Show**. Notes that the previous text points are dimmed as each subsequent line appears.

9.10. Rehearse Timings:

The Rehearse Timing Tool allows you to practice giving an on-screen presentation, regardless of any slide advance timings you may have set previously. The tool sets time not only for the entire presentation, but also each individual slide, as you rehearse. Each time you advance the slide, the time is recorded. You must be in Slide Sorter View to access the Rehearse Timing Tool.

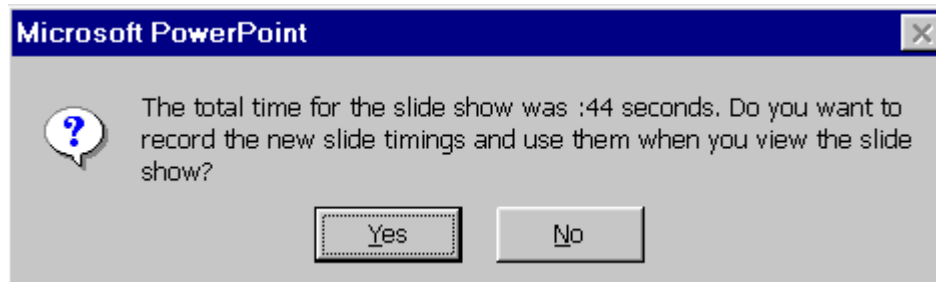


Sample Recorded Rehearsal Timings



To use the Rehearse Timing Tool, click the Rehearse Timing Tool.

When you have completed rehearsing, PowerPoint displays the total time for the rehearsed presentation and asks if you want to record the new timings. If you click "Yes", these new timings will replace any slide advance timings you may have previously set for the presentation.



Sample New Timings Dialog Box

Before you print your slide presentation, make sure the slide size and orientation is correct. Also, decide whether you want to print in color or black and white.

Change these settings using the Page Setup and Print commands on the File menu.

Slide Setup

The Slide Setup dialog box has controls for slide size and orientation. Choose the appropriate slide size and orientation before you do significant formatting in your presentation. Changing slide orientation from portrait to landscape or vice versa may require editing and reformatting slides.

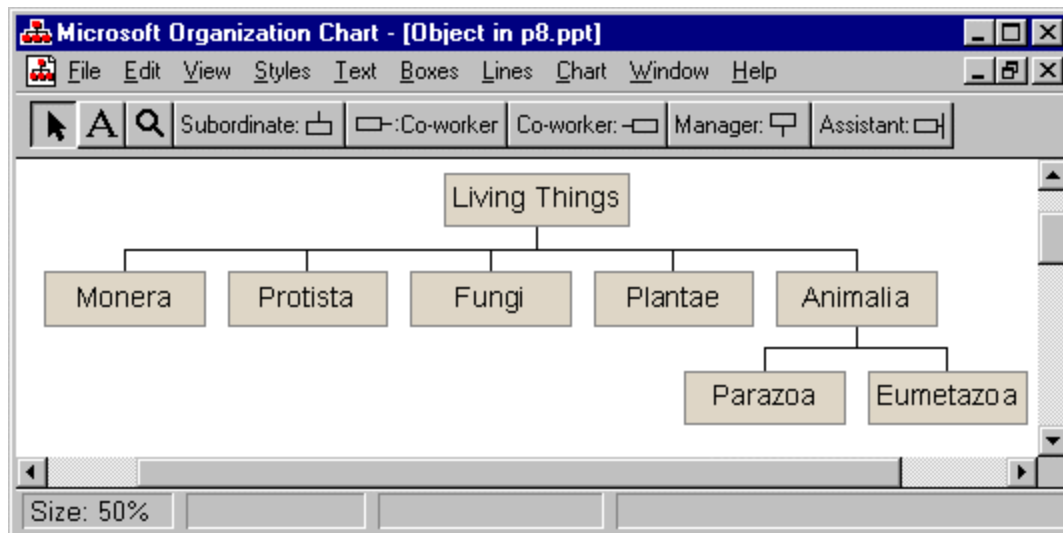
Printing

To print your slide presentation, choose Print from the File menu, or click the Print button on the toolbar. In the Print dialog box you can choose how many copies you want to print. Use the Print What: drop-down list to select whether you want to print just the slides, notes, or handouts with multiple slides per page.


9.12. Charts:

To insert an *organization chart* (depicts hierarchical relationships):

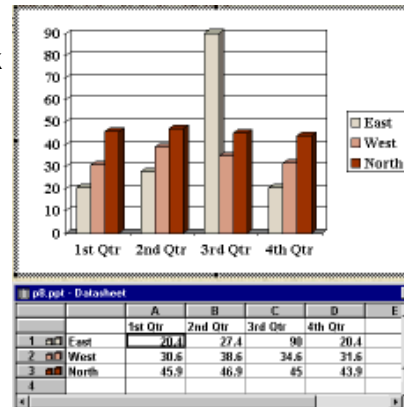
- First display the slide on which you want the chart to appear.
- On the **Insert** menu, point to **Picture**, and then click **Organization Chart**.
- A separate **Microsoft Organization Chart** window appears.
- Use the buttons at the top of the Organization Chart window to add text boxes to your chart (above, below, or beside existing boxes).
- Click your mouse inside a box to edit the text inside of it.
- As shown in the example below, you can use this type of chart for almost any type of information you want, not just managers, workers, and other positions within a company.



To create a *bar or pie chart* (or similar chart based on numbers in a datasheet):

- First display the slide on which you want the chart to appear.
- On the **Insert** menu, click **Chart** - or click the **Insert Chart** button on the PowerPoint toolbar. 
- A sample chart and datasheet (spreadsheet) appear on the screen as shown below.

- To replace the sample data, click a cell on the datasheet, and then type the information you want.
- To change the chart type, on the **Chart** menu, click **Chart Type**. In the **Chart Type** window, which appears, click on the desired Chart type and sub-type and then click the **OK** button.
- To return to PowerPoint, where you can move and size the chart, click outside the chart on the PowerPoint slide.
- If you wish to re-edit the chart, double-click the chart on the PowerPoint slide.



9.13. Summary:

At the end of this lesson you have learnt how to insert and format text, working with the pictures and inserting images into a slide. The concept of Slide transition, Slide animation, Rehearse timings allows you to automatically move to next slide without manual interaction. A layman can also know more about the data results with the help of charts.

9.14. Technical Terms:

Hyperlink: Hyperlinks cause the selection of an object to result in a move to a new location or the performance of an action.

Slide Transition: Slide Transition determine the effects applied when you move from one slide to another during an on-screen presentation.

Slide Show Animation: Slide Show Animation can aid in focusing attention during a presentation and make your slide show more interesting.

Rehearse Timings: The Rehearse Timing Tool allows you to practice giving an on-screen presentation, regardless of any slide advance timings you may have set previously.

Slide Setup: The Slide Setup controls for slide size and orientation.

9.15. Model Questions:

1. Explain the procedure to draw the objects in PowerPoint?
2. Define Hyperlink? Explain the procedure to create Hyperlinks in PowerPoint?
3. What is a Slide Miniature?
4. Define Rehearse Timing? Explain the procedure to set the rehearse timings to slides?

9.16. References:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, ' The ABC'S Of Microsoft Office. ' BPB Publications, New Delhi

MS OFFICE 2000 Complete, BPB Publications, New Delhi

- C.V.P.R.PRASAD, M.C.A.,

Lecturer, Dept.Of Computer Science,

JKC College, GUNTUR.

Lesson– 10

MICROSOFT ACCESS

10.0 Objective:

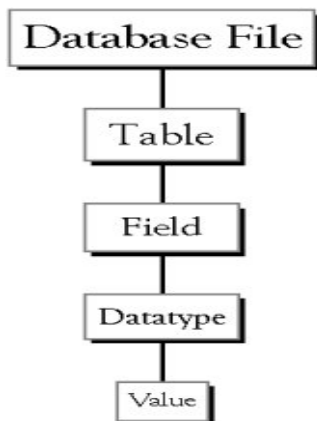
This lesson provides an introduction to the fundamentals of Microsoft Access. It covers starting MS-Access, creating Data bases, Tables, Queries, Forms, Reports, and Relational Data Bases.

Structure of the Lesson:

- 10.1. Microsoft Access Introduction
- 10.2. Starting Microsoft Access
- 10.3. Creating and opening Data bases
- 10.4. Tables
- 10.5. Query
- 10.6. Relational Data Bases
- 10.7. Forms
- 10.8. Reports
- 10.9. Summary
- 10.10. Technical Terms
- 10.11. Model Questions
- 10.12. References

10.1. Microsoft Access:

Microsoft Access is a powerful program to create and manage your databases. It has many built in features to assist you in constructing and viewing your information. Access is much more involved and is a more genuine database application than other programs such as Microsoft Works.



Database File: This is your main file that encompasses the entire database and that is saved to your hard-drive or floppy disk.

Example studentDatabase.mdb

Table: A table is a collection of data about a specific topic. There can be multiple tables in a database.

Example #1) Students

Example #2) Teachers

Field: Fields are the different categories within a Table. Tables usually contain multiple fields.

Example #1) Student LastName

Example #2) Student FirstName

Data types: Data types are the properties of each field. A field only has 1 data type.

10.2. Starting Microsoft Access:

Starting Microsoft access in two ways.

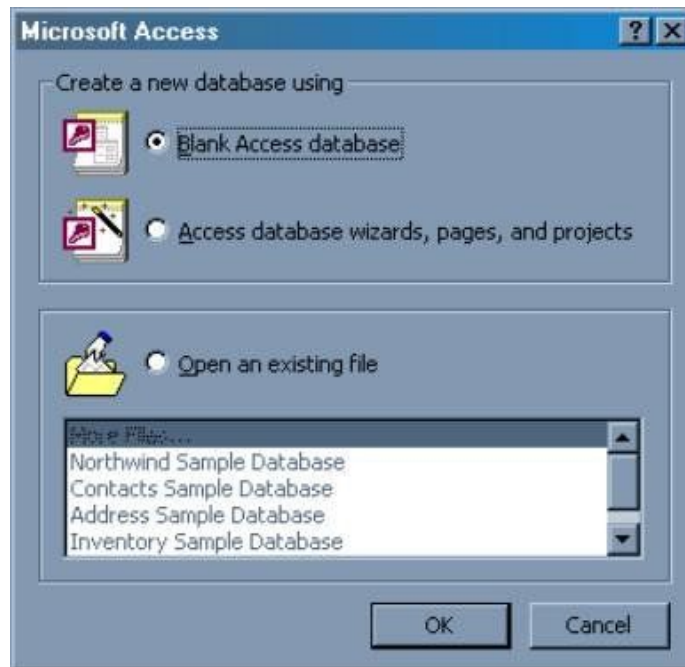
1. Double click on the Microsoft Access icon on the desktop.



2. Click on Start --> Programs --> Microsoft Access

10.3. Creating New and Opening Existing Databases:

If you start MS Access a window will appear as below.



The above Window gives you the option to:

- Create a New Database from scratch
- Use the wizard to create a New Database

- Open an existing database
- The white box gives you the most recent databases you have used. If you do not see the one you had created, choose the More Files option and hit OK. Otherwise choose the database you had previously used and click OK.

Create a database using the Database Wizard

1. When Microsoft Access first starts up, a dialog box is automatically displayed with options to create a new database or open an existing one. If this dialog box is displayed, click **Access Database Wizards, pages, and projects** and then click **OK**.

If you have already opened a database or closed the dialog box that displays when Microsoft Access starts up, click **New Database** on the toolbar.

2. On the **Databases** tab, double-click the icon for the kind of database you want to create.
3. Specify a name and location for the database.
4. Click **Create** to start defining your new database

Create a database without using the Database Wizard

1. When Microsoft Access first starts up, a dialog box is automatically displayed with options to create a new database or open an existing one. If this dialog box is displayed, click **Blank Access Database**, and then click **OK**.

If you have already opened a database or closed the dialog box that displays when Microsoft Access starts up, click **New Database** on the toolbar, and then double-click the **Blank Database** icon on the **General** tab.

2. Specify a name and location for the database and click **Create**. (Below is the screen that shows up following this step).

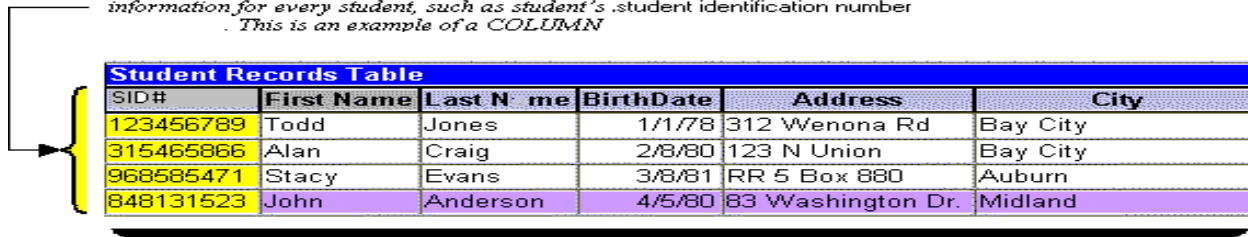


10.4. Tables:

A table is a collection of data about a specific topic, such as students or contacts. Using a separate table for each topic means that you store that data only once, which makes your database more efficient, and reduces data-entry errors.

Tables organize data into columns (called **fields**) and rows (called **records**).

Each field in the Student Records table contains the same type of information for every student, such as student's student identification number. This is an example of a COLUMN.



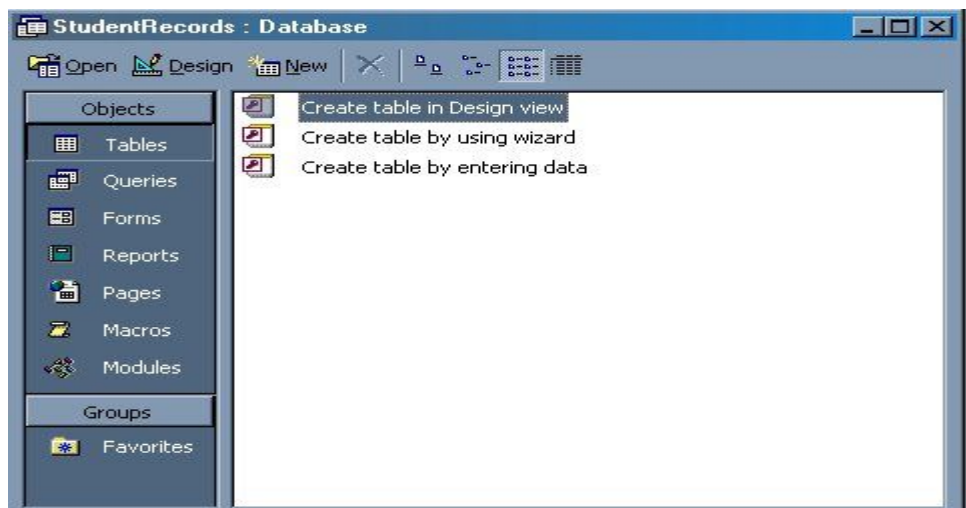
The diagram shows a table titled "Student Records Table" with six columns: SID#, First Name, Last Name, BirthDate, Address, and City. The first row is highlighted in yellow, the second in green, the third in purple, and the fourth in blue. A yellow bracket on the left side of the table groups the columns, and a blue bracket at the bottom groups the rows.

SID#	First Name	Last Name	BirthDate	Address	City
123456789	Todd	Jones	1/1/78	312 Wenona Rd	Bay City
315465866	Alan	Craig	2/8/80	123 N Union	Bay City
968585471	Stacy	Evans	3/8/81	RR 5 Box 880	Auburn
848131523	John	Anderson	4/5/80	83 Washington Dr.	Midland

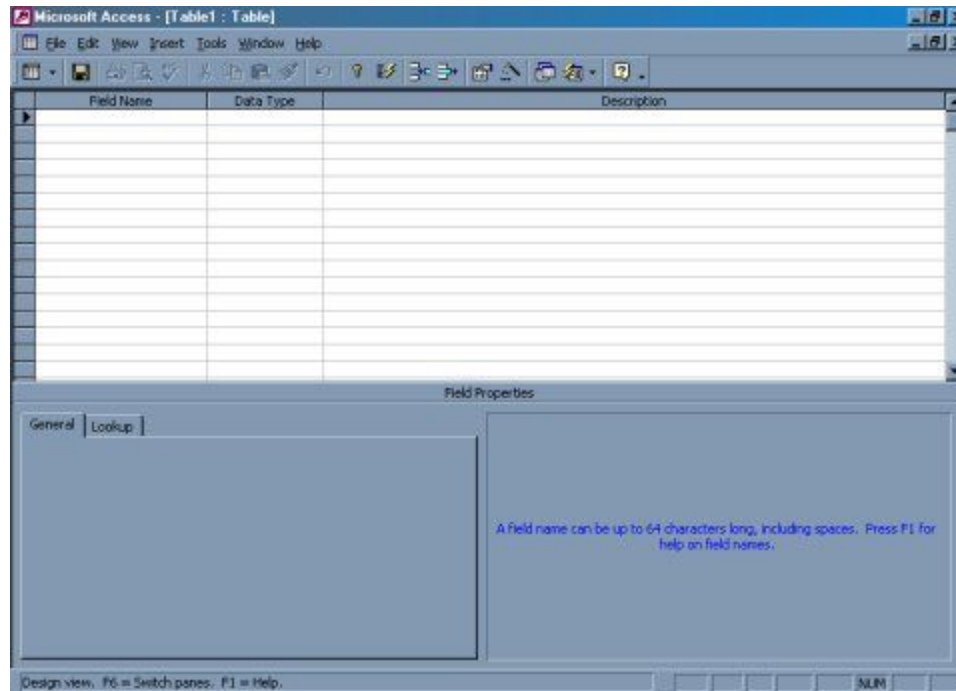
Each record in a Student Records table contains all of the information about one student, such as their First Name, Last Name, Birthday, Address, and City, etc... This is an example of a ROW.

Create a Table from scratch in Design view

1. If you haven't already done so, switch to the Database Window You can press F11 to switch to the Database window from any other window.



2. Double-Click on "**Create table in Design view**".
(*DESIGN VIEW*)



3. Define each of the fields in your table.
 - Under the Field Name column, enter the categories of your table.
 - Under Data Type column, enter the type you want for your categories.
 - The attribute of a variable or field determines the kind of data it can hold. For example, in a Microsoft Access database, the **Text** and **Memo** field data types allow the field to store either text or numbers, but the **Number** data type will allow the field to store numbers only. Number data type fields store numerical data that will be used in mathematical calculations. Use the **Currency** data type to display or calculate currency values. Other data types are **Date/Time**, **Yes/No**, **Auto Number**, and **OLE object** (Picture).
 - Under the Description column, enter the text that describes what your field is. (This field is optional).
 - For example

Field Name	Data Type	Description
stdid	Text	stdid Uniquely identifies a student
First Name	Text	Student's First Name
Last Name	Text	Student's Last Name
BirthDate	Date/Time	Student's Birthdate
Address	Text	Students Address
City	Text	City student resides in
State	Text	State student resides in
Zip	Text	Zip Code student resides in
Phone	Text	Student's home phone number

Primary Key

- One or more fields (columns) whose value or values uniquely identify each record in a table. A primary key does not allow Null values and must always have a unique value. A primary key is used to relate a table to foreign keys in other tables.
- In the above example, the **Stdid #** field the primary key, meaning that every student has a student identification number.
 - To do this, simply select the **Stdid #** field and select the primary key button.
 - After you do this, Save the table

Switching Views

- Views are two types. They are Design view and Datasheet view. You can select the views from view menu from menu bar.

Datasheet View	Design View
<div data-bbox="456 1661 529 1709" data-label="Image"> </div> <p>Displays the view, which allows you to enter raw data into your database table.</p>	<div data-bbox="997 1661 1070 1709" data-label="Image"> </div> <p>Displays the view, which allows you to enter fields, data-types, and descriptions into your database table.</p>

Entering Data

- Click on the Datasheet View and simply start "chugging" away by entering the data into each field.



Manipulating Data

- **Adding a new row**
 - Simply drop down to a new line and enter the information.
- **Updating a record**
 - Simply select the record and field you want to update, and change its data with new contents
- **Deleting a record**
 - Simply select the entire row and hit the Delete Key on the keyboard

10.5. Query:

Queries are questions about your database. You can design a query manually, specifying all of the tables and fields you wish to display on your screen, or you can use a Query Wizard to create simple displays in predefined formats.

QUERY WIZARD

Access provides several query formats

- *simple* - looks like the Table datasheet view
- *crosstab* – is similar to a spreadsheet with summary information inserted into the intersection of each row and column
- *find duplicates* – lists the records which have duplicate values in any specified field
- *find unmatched* – compares two tables and identifies records, which do not have a match in a related table.

Using Query Wizard

From Database window:

- Click Query Tab
- Click New
- Select Query type from list (OK)
- Follow prompts:
 - select Tables / Queries (Next)
 - select Fields (Next)
 - create a Title
 - click Finish.

In general, there are two types of query: Select and Action. A select query gathers, collates and presents information in usable forms. An action query makes changes in specified records of an existing table, or creates a new table. In this course, we will concentrate on select queries only.

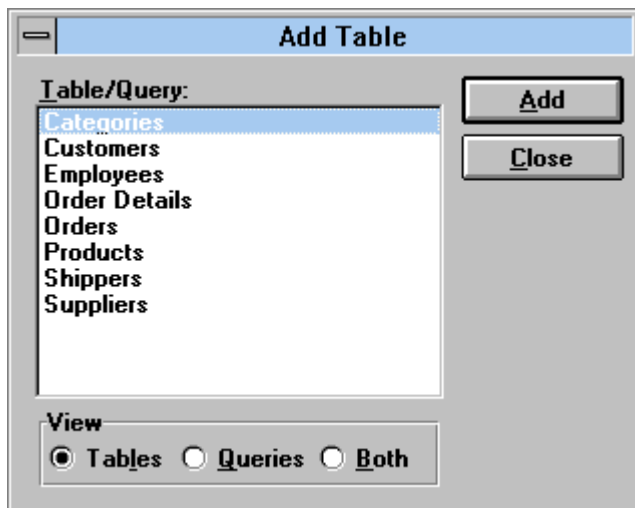
Sometimes, information is scattered in many tables. For example, customers' information is stored in **Customers** table while their ordering information is stored in **Orders** table. In order to retrieve information from more than one table, queries are used. Moreover, queries are capable of performing the functions of filters, namely adding selection criteria and sorting instructions. In addition, queries allow calculation of new fields.

In contrast to a filter, a query is **reusable**. When a table is closed, the selection criteria and/or sorting instructions will be lost. Therefore, in order to retrieve the same records again, the filter has to be recreated. However, a query allows the selection criteria and/or sorting instructions to be saved and reused.

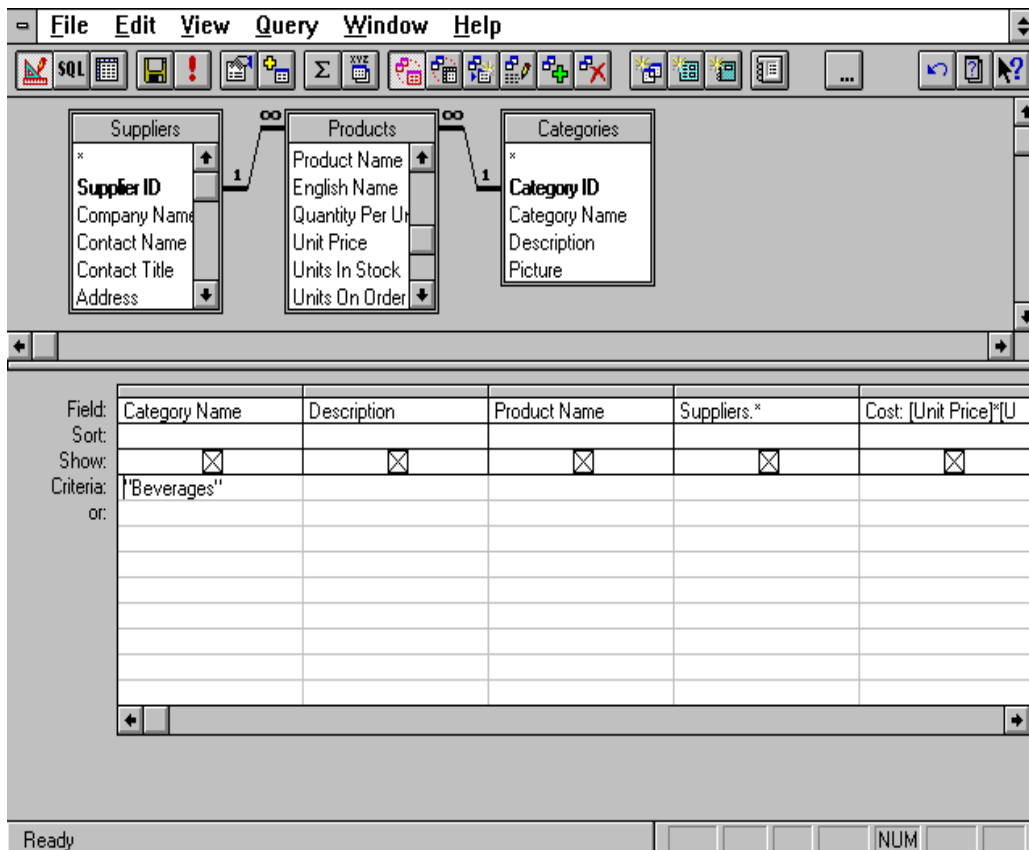
Create a Query

Find all the products under the category Beverages and the total costs for each of these products. In addition, you have to find the suppliers' information of these products.

1. Click the **Query** button and then the **New** button in the Database window.
2. Click **New Query** in the resulting dialog box to open the design window for a new query.
3. From the **Add Table** dialog box, select **Categories**, **Products** and **Suppliers** tables. Click **Close** to close the **Add Table** dialog box.



4. Drag the **Category Name** and **Description** fields from the **Categories** field list to the first and second column in the grid. Then drag the **Product Name** fields from the **Products** field list to the third column. Finally, drag the * field from the **Suppliers** field list to the fourth column.
5. In the fifth column of the grid, type **Cost: [Unit Price]*[Units in Stock]*0.7**.
6. Select the **Criteria** row in the **Category Name** column and enter **Beverages**.
7. Run the query by clicking the **Run** button (!) in the tool bar.



8. To print the result, choose **Print** command from the **File** menu.
9. To save the query, choose **Save** command from the **File** menu.

10.6. Relationships:

After you have set up multiple tables in your Microsoft Access database, you need a way of telling Access how to bring that information back together again. The first step in this process is to define relationships between your tables. After you have done that, you can create queries, forms, and reports to display information from several tables at once.

A relationship works by matching data in key fields - usually a field with the same name in both tables. In most cases, these matching fields are the primary key from one table, which provides a unique identifier for each record, and a foreign key in the other table. For example, teachers can be associated with the students they're responsible for by creating a relationship between the teacher's table and the student's table using the TeacherID fields. Having met the criteria above, follow these steps for creating relationships between tables.

1. In the database window view, at the top, click on Tools ---> Relationships

2. Select the Tables you want to link together, by clicking on them and selecting the Add Button
3. Drag the primary key of the Parent table (Teacher in this case), and drop it into the same field in the Child table (Student in this case.)



4. Select **Enforce Referential Integrity**



- When the Cascade Update Related Fields check box is set, changing a primary key value in the primary table automatically updates the matching value in all related records.
 - When the Cascade Delete Related Records check box is set, deleting a record in the primary table deletes any related records in the related table
5. Click Create and Save the Relationship

10.7. Forms:

A form is nothing more than a graphical representation of a table. You can add, update, and delete records in your table by using a form.

A form is very good to use when you have numerous fields in a table. This way you can see all the fields in one screen, whereas if you were in the table view (datasheet) you would have to keep scrolling to get the field you desire.

Create a Form using the Wizard

It is a very good idea to create a form using the wizard, unless you are an advanced user and know what you are doing. Microsoft Access does a very good job of creating a form using the wizard. The following steps are needed to create a basic form:

1. Switch to the Database Window. You can do this by pressing F11 on the keyboard.
2. Click on the **Forms** button under **Objects** on the left side of screen.
3. Double click on **Create Form Using Wizard**
4. On the next screen select the fields you want to view on your form. Most of the time you would select all of them.
5. Click Next.
6. Select the layout you wish.
7. Click Next.
8. Select the style you desire.
9. Click Next.
10. Give your form a name, and select **Open the Form and enter information**.
11. Select **Finish**.
12. You should see your form. To adjust the design of your form, simply hit the design button (same as with the tables), and adjust your form accordingly.

10.8. Reports:

A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.

Create a Report using the Wizard

As with the Form, it is a very good idea to create a report using the wizard, unless you are an advanced user. Microsoft Access does a very good job using the wizard to create reports.

1. Switch to the Database Window. You can do this by pressing F11 on the keyboard.
2. Click on the **Reports** button under **Objects** on the left side of screen.
3. Double click on **Create Report Using Wizard**.
4. On the next screen select the fields you want to view on your form. Most of the time you would select all of them.
5. Click Next.
6. Select if you would like to group your files. Keep repeating this step for as many groupings as you would like.
7. Click Next.
8. Select the layout and the paper orientation you desire.
9. Click Next.
10. Select the style you desire.
11. Click Next.
12. Give your report a name, and select **Preview the Report**.
13. Select **Finish**.
14. You should see your report. To adjust the design of your report, simply hit the design button, and adjust your report accordingly.

10.9. Summary:

At the end of this lesson you have learnt what is a database, which contains Tables, Queries, Forms and Reports. Table is a collection of data to a specific content which contains some rows and columns. You have learnt the procedure how to create tables and Queries. Queries are questionnaires to a database. A form is nothing more than a graphical representation of a table. You can add, update, and delete records in your table by using a form. Reports are used to present your data in a printed format.

10.11. Technical Terms:

Database: A database is a collection of information stored in a computer in a systematic way, such that a computer program can consult it to answer questions.

Table: A table is a collection of data about a specific topic, such as students or contacts.

Primary Key: Primary Key is a field or whose value or values uniquely identify each record in a table.

Query: Queries are questionnaires to a database.

Form: A form is nothing more than a graphical representation of a table.

Relationship: Relationship is used to bring the information back together again.

Report: Reports are used to present your data in a printed format.

10.12. Model Questions:

1. Explain about the creation of databases in MS-Access.
2. Explain the procedure to create a Table with an example?
3. Define a Query? Explain the procedure to create query-using relationships?
4. Explain different data types in MS-Access.
5. What is a Primary Key?
6. What is a Form? Write the procedure to create a form in different ways?

10.13. References:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, ' The ABC'S Of Microsoft Office. ' BPB Publications, New Delhi

MS OFFICE 2000 Complete, BPB Publications, New Delhi

Y.SURESH BABU, M.Com., M.C.A.,
Lecturer,
Dept. Of Computer Science,
JKC College,
GUNTUR.

Lesson- 11**MICROSOFT EXCEL****Objectives:**

This lesson introduces the basic features of Microsoft Excel, a spreadsheet program, entering and editing data, formatting, formulas, page setup, and printing spreadsheets.

Structure of the Lesson:

- 11.1. Excel Features
- 11.2. Starting Microsoft Excel
- 11.3. Formula Bar
- 11.4. Working with Excel
- 11.5. Entering and Editing Data
- 11.6. Row and Column Headings
- 11.7. Auto Fill
- 11.8. Custom Fill Lists
- 11.9. Row or Column Size
- 11.10. Inserting Rows and Columns
- 11.11. References
- 11.12. Charts
- 11.13. Summary
- 11.14. Technical Terms
- 11.15. Model Questions
- 11.16. References

11.1. Features of Excel:

Answer Wizard: Like word, excel has a new feature called answer wizard. It uses “IntelliSense” to determine what kind of work you need, based upon your questions.

Autocalc: When we selected a group of numbers, the sum automatically appears on the status area. We can see average, count the numbers by right clicking on the status area.

Auto Complete: Auto complete will try to figure out what you intend to type, based upon the entries you have already made.

Auto Correct: Excel can automatically correct the mistakes. It will not allow two consecutive initial caps, capitalizing names of days and so on. You can create your own auto correct entries.

Drag And Drop: Like other office features we can drag and drop the group of cells using mouse. We can drag the specified portion of from one worksheet to other.

Cell tips and Scroll Tips: When you touch the scroll bar it will display the row and column number that you are heading for. This is great for large worksheets. You can use Cell tips to read notes you’ve made for yourself in cells.

Data Map: It is having a data map feature. We can make maps of different regions in different countries. This can be quite impressive in presentations. This can be helping you in our business matters.

Easier retrieval of Documents: It has greatly streamlined file management. It can quickly find worksheets and you can easily delete or rename files from the open dialog box.

Number Formatting: It is easy to format the numbers with Excel’s new number-formatting feature. There are many types of formats here like telephone numbers, dollar amounts, and dates.

Templates and Wizards: A variety of templates (predefined) formats available with excel for use in business organizations. We can have a template wizard to link database with worksheets. If we made changes in the worksheet, automatically database updated.

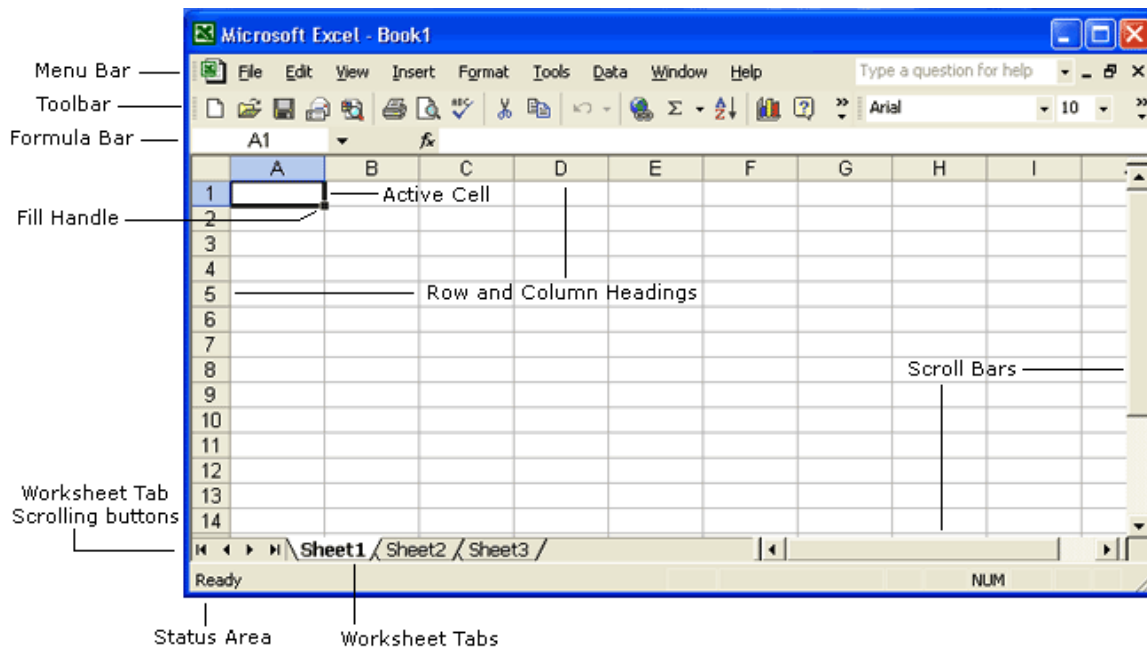
11.2. Starting Excel:

On the PC, select Start, Programs, and Microsoft Excel from the Start list.

Components of the Excel Window

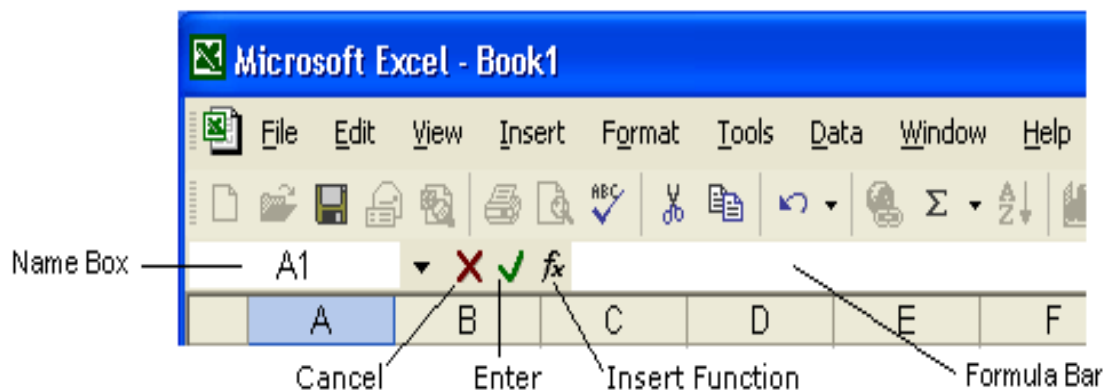
Besides the usual window components (close box, title bar, scroll bars, etc.), an Excel window has several unique elements identified in the figure below:

11.3. Formula Bar:



11.3. Formula Bar:

In addition to the other toolbars the formula bar is located under the formatting toolbar at the top of the Excel worksheet. Use the formula bar to enter and edit worksheet data.



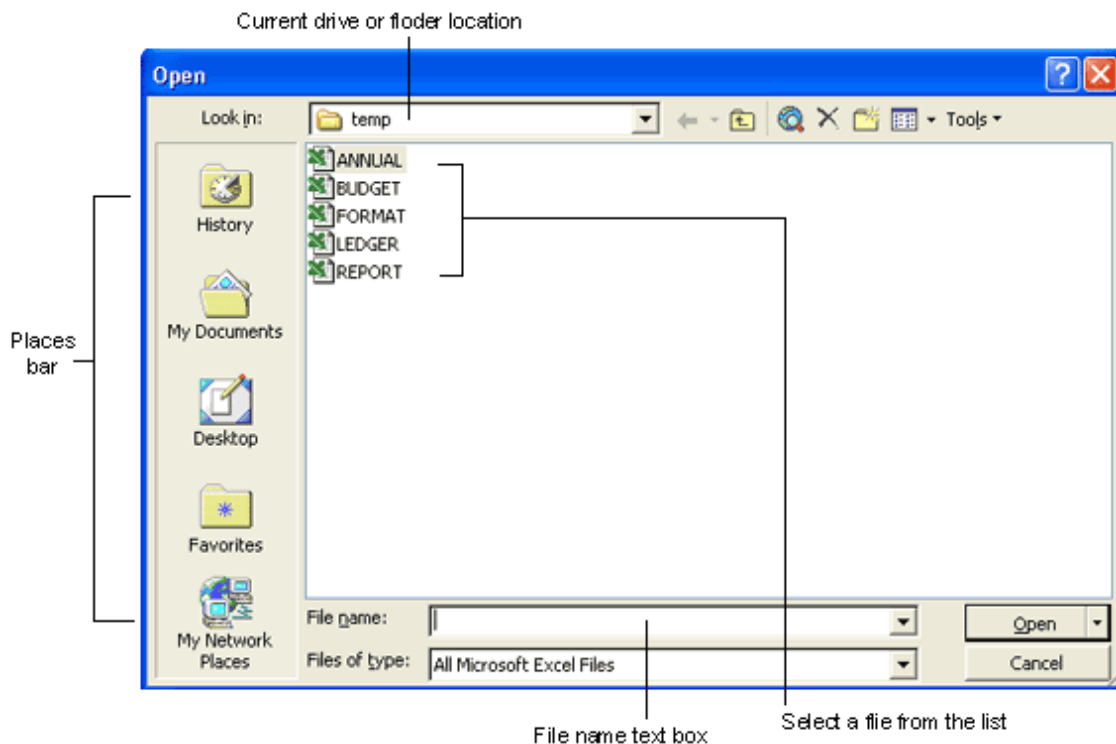
The contents of the active cell always appear in the formula bar. When you click the mouse in the formula bar, an X and a check mark appear. You can click the check icon to confirm and complete editing, or the X to abandon editing.

11.4. Working with Excel:

Opening and Closing worksheet

To open an existing Excel workbook, choose **Open** from the **File menu** or click the **Open tool** on the **toolbar**. Excel will display the **Open dialog box**, which is similar in appearance and function to the **Save As** dialog box. It contains a File Name: box for you to type a file name.

To open a file, select the appropriate location by clicking on the down pointing arrow next to the **Look In:** selection box. Excel will display all Excel files found there. Select the file you wish to open and click Open to open the file.



Saving a worksheet

To save an untitled Excel workbook, from the File menu choose **Save As** or click the **Save** button on the toolbar. The **Save As** dialog box contains a text box for you to type a specific filename, a "Save File as Type:" box to save your workbook in a different format that other programs can read, a selection box to designate the folder and drive in which to save the file, and a "Places Bar" that offers other locations to save your file.

Creating a New Workbook

If you are already in Excel and you want to create a new workbook, choose **New** from the File menu or click the New tool on the toolbar.

Managing Multiple Workbooks

Excel enables you to have more than one workbook open simultaneously. Each open workbook appears on the **Window menu**. The document with the check next to it is the active document. To switch to another document, simply choose that document from the **Window** menu. To navigate between worksheets within a workbook, click the worksheet tab you want to activate. Double-click a worksheet tab to change its name.

11.5. Entering and Editing Data:

Entering Data

You can enter text, numbers and dates in an Excel worksheet. In Excel's terminology, numbers and dates are called **values** and text is referred to as a **label**. To enter data of any type, click on the cell you want to contain that data, and then type the information you want in the cell. When you begin typing, your data also appears in the formula bar. When you have finished typing the data for the active cell, press the Return or Enter key.

Editing Data

The easiest way to edit the contents of a cell is to select the cell and then retype the entry. The new entry replaces the old contents. For example, to change the number in cell C8 to 123, select cell C8, type 123 and press Return. This method works well with numbers, but is more difficult when editing long text labels or formulas.

The formula bar gives you more flexibility while editing. When the mouse pointer moves into the formula bar, it changes shape to an I-beam, signifying that you can enter or edit text. Text in the active cell appears in the formula bar and you can edit it there. Use the mouse to select the text you want to change in the formula bar and then type the new text. Excel automatically replaces what is selected. Don't forget to press the Return key when you finish editing a cell.

You can also edit labels and values directly in cells. Double-click on the cell you want to edit. This puts an insertion point in the cell. Edit the contents of that cell.

Moving the Active Cell

Cell selection and movement around the worksheet are similar operations in Excel. To select a given cell or make it active, simply click on that cell. Use the mouse or the arrow keys to move around the worksheet. For example, if you press the right arrow key twice you move two cells to the right.

Name box

The Name box displays the reference of the selected cells.

11.6. Row and Column Headings:

Letters and numbers identify the rows and columns on an Excel worksheet. The intersection of a row and a column is called a cell. Use row and column headings to specify a cell's reference. For example, the cell located where column C and row 8 intersect is called C8.

Active cell

The active cell has a dark border around it to indicate your position in the worksheet. All text and numbers that you type are inserted into the active cell. Click the mouse on a cell to make it active.

Fill handle

The lower right corner of the active cell has a small box called a Fill Handle. Your mouse changes to a cross hair when you are on the Fill Handle. The Fill Handle helps you copy data and create series of information. For example, if you type Monday in the active cell and then drag the Fill Handle over four cells, Excel automatically inserts Tuesday, Wednesday, Thursday, Friday and Saturday.

Worksheet tabs

An Excel workbook consists of 3 worksheets by default. Use the worksheet tabs at the bottom of the screen to navigate between worksheets within a workbook.

Go To Command

A quick way to move a large distance on a worksheet is with the Go To command on the Edit menu. When you select the Go To command, a dialog box prompts you to identify the cell. Enter the cell reference and click OK or press Return.

Clearing Cell Contents

To clear the contents of a cell choose **Clear** from the **Edit** menu. Then, select what you want to clear from the cell: All, Formats, Contents, or Notes. Most frequently you will want to clear the Contents of a cell. Pressing the **Delete** key also clears the contents of cells.

Undoing Mistakes

If you make a terrible mistake and you accidentally delete important data. Use the **Undo** command on the **Edit** menu or the Undo tool to correct the mistake.

Copying and Moving Data

Copy selected data from one cell to another with the **Copy** and **Paste** commands, or with the Drag and Drop procedure. If you want to move data use the **Cut** and **Paste** commands, or the Drag and Drop procedure.

Drag and Drop

This procedure does not involve the Clipboard and works best when moving data a short distance. To move cells, position the mouse on the cell borders. Wait until the mouse pointer changes to a left-pointing arrow then press and drag the data to the new location.

11.7. Auto Fill:

A series of numbers, month's names or week names can be filled automatically in Excel. For doing this simply type the first word in a cell then place the mouse pointer on the fill handle and then drag it either horizontally or vertically so that Excel fills the data automatically.

11.8. Custom Fill Lists:

Excel providing a facility to create our own fill lists we can create our college courses, names, a series of your colleague's name. These are the steps to create custom lists.

- Select Tools/Options.
- Click on custom lists.
- Click on New list on custom area.
- Type your list of entries in list entries area.
- Select add button.
- Your new list will be created.

11.9. Row or Column Size:

The height of a row or the width of a column can be changed to any desired value.

Row height

Select the row, which you want to increase the height, select Row from format menu. Specify the value. Then automatically the height of the Row increases.

Or

Select the row, then place the mouse pointer on the bottom borders of the row then the mouse pointer changes. Press left mouse button and drag the mouse downwards until the desired row height reaches.

Column Height

Select the Column, which you want to increase the width. Select Column from format menu. Specify the value. Then automatically the width of the column increases.

Or

Select the column, then place the mouse pointer on the top borders of the column then the mouse pointer changes. Press left mouse button and drag the mouse downwards until the desired column width reaches.

11.10. Inserting Rows and Columns:

Select Insert menu from menu bar and select row option. Then automatically the row will be inserted. Similarly select Insert menu from menu bar and select column option. Then automatically a column will be inserted.

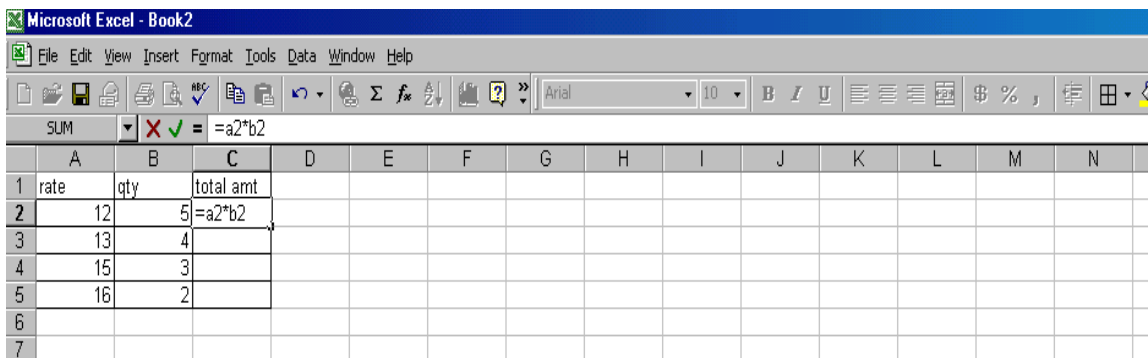
11.11. References:

About cell and range references

A reference identifies a cell or a range of cells on a worksheet and tells Microsoft Excel where to look for the values or data you want to use in a formula. With references, you can use data contained in different parts of a worksheet in one formula or use the value from one cell in several formulas. You can also refer to cells on other sheets in the same workbook.

Relative reference

Relative cell references are references to cells relative to the position of the formula. When you create a formula, references to cells or ranges are usually based on their position relative to the cell that contains the formula. In the following example



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Book2". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar contains various icons for file operations, editing, and formatting. The formula bar shows the formula $=a2*b2$. The worksheet has columns labeled A through N and rows numbered 1 through 7. The data in the worksheet is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	rate	qty	total amt											
2	12	5	$=a2*b2$											
3	13	4												
4	15	3												
5	16	2												
6														
7														

cell c2 contains the formula $=A2*B2$; Microsoft Excel finds the value one cell above and one cell to the left of c2. This is known as a relative reference. When you copy a formula that uses relative references, Excel automatically adjusts the references in the pasted formula to refer to different cells relative to the position of the formula. The formula in cell c2 $=a2*b2$, the formula has been copied to cell C3. Excel has adjusted the formula in cell C3 to $=A3*B3$, which refers to the cell i.e., one cell above and to the left of cell C3.

Absolute reference

Absolute references, which are cell references that always refer to cells in a specific location. If a dollar sign precedes the letter and/or number, such as $\$A\1 , the column and/or row reference is absolute. Relative references automatically adjust when you copy them, but absolute references do not.

For example:

	A	B	C	D	E	F	G	H
1	rate	qty	total amt			vat		
2	12	5	=a2+b2+\$f\$2			10		
3	13	4	52					
4	15	3	45					
5	16	2	32					
6								
7								
8								
9								
10								
11								
12								
13								

The total amount is calculated by the formula $\text{rate} \times \text{qty} + \text{vat}$. The fixed vat price has to be applied to all the transactions. Here we are applying absolute reference for the cell f2. i.e., \$F\$2.

Named reference

You can assign names to cells or ranges of cells. When you do that, you can use the names in your formulas. For example, if you had a column named QTY and a column named Price, you could create a formula like $\text{= QTY} * \text{Price}$.

11.12. Charts:

Excel has features for preparing charts for the selected data in a worksheet. A chart can be drawn in the same worksheet in which data is available or can be drawn as a separate chart sheet. Chart sheets have again tabs in the tab area, which operates as worksheets tabs. All charts in Excel are available in chart wizard so that a chart can be prepared very easily by using this special chart wizard.

The Best Chart Types

There are a lot of different chart types out there, and it can be confusing to work out just which chart type you need for a particular set of data. Some of the most common types of chart and their uses are explained below. The definitions of the charts remain the same no matter what operating system you use, or what spreadsheet program.

Pie Chart Type

Pie charts should only be used if you want to show proportions. This chart type emphasizes the relationship between the whole and each part of that whole. A good example of this would be to display data on what spreadsheet program is preferred by 100 of the top spreadsheet experts.

Column/Bar Chart Type

The bar chart is undoubtedly the king of all chart types. Its simplicity makes it so useful for a range of data types. It is the best chart type to use if you need to compare values.

Scatter Chart Type

Scatter charts (also known as XY-Scatter charts) can be used when you need to display two related sets of data on a chart, particularly if you want to make predictions based on the data.

Line Chart Type

Line charts should generally only be used where there is a period of time involved. These charts are perfect for displaying trends. For example, if you want to chart the increase in use of open source spreadsheets during a 5-year period, the line chart would be an excellent choice for your chart type.

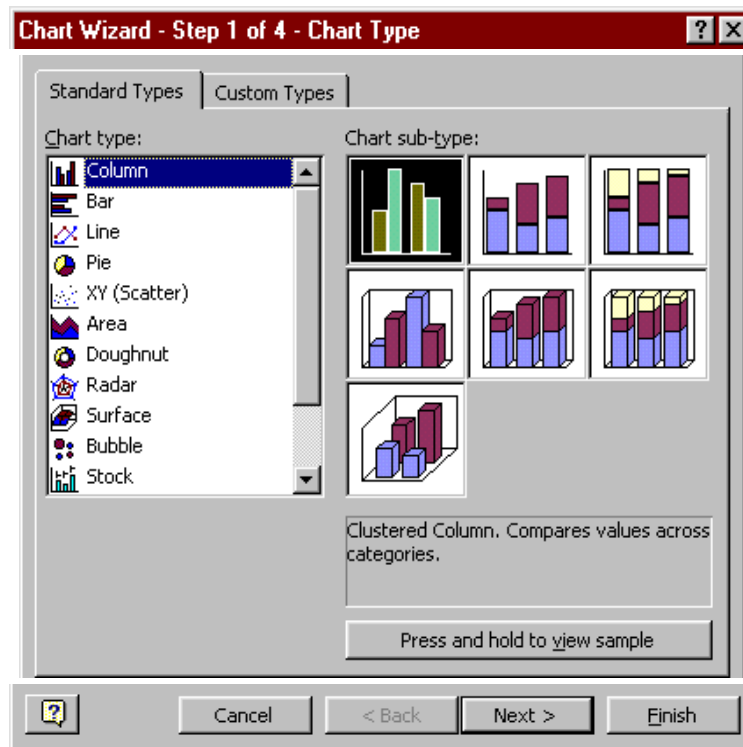
The first step to making our chart is to select the range of data to be charted.

	A	B	C	D	E	F	G
1							
2			Apples	Oranges	Grapes	Bananas	
3		January	13	8	8	13	
4		February	8	11	3	7	
5		March	4	7	7	15	
6		April	8	8	8	8	
7		May	7	11	11	4	
8		June	12	3	12	15	
9		July	9	7	11	11	
10		August	4	14	15	9	
11		September	7	7	8	13	
12		October	10	9	5	15	
13		November	8	15	15	6	
14		December	5	13	6	3	
15							
16							
17							

The **Chart Wizard** can be started by using the **Chart Wizard** button on the Standard command bar (left) or from the **Chart...** button on the **Insert menu**. The data can be selected prior to running the chart wizard, filled in while running the chart wizard, or added to the chart later. The Chart Wizard runs through a series of steps that help you set up your chart.

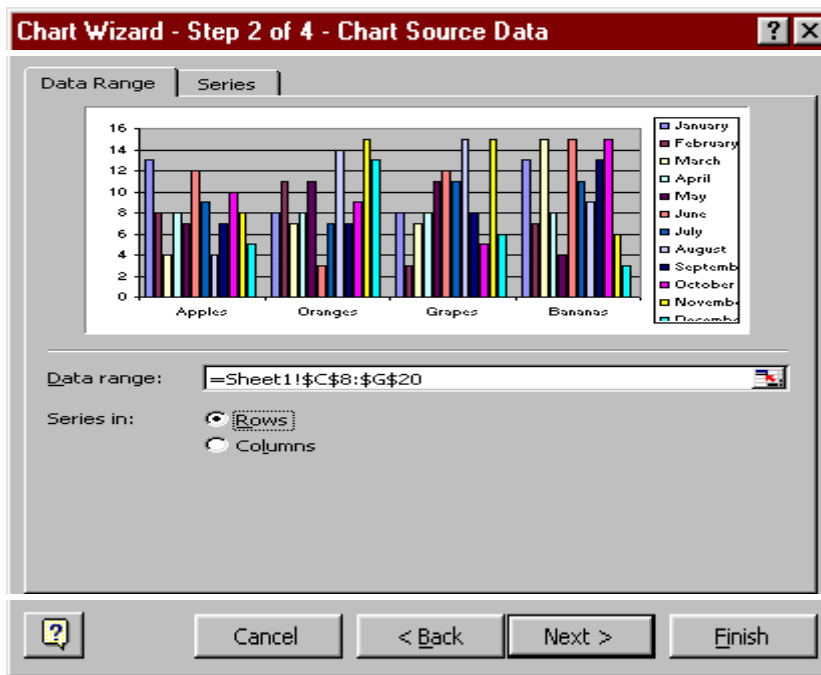
Step 1 - Select a Chart Type


The Standard Types tab of Step 1 of the Chart Wizard shows a list of chart types on the left side, and several chart sub-types on the right. The Custom Types tab has a number of built-in custom types of charts, including several combination charts. You can add your favorite custom chart types to the User-Defined list.



Step 2 - Source Data

The Data Range tab gives you a preview of the chart style you selected in Step 1, indicating where the chart's source data range is, and allowing you to plot by row or by column. The first dialog box below shows a column chart, plotted by rows. The second shows an area chart plotted by columns.




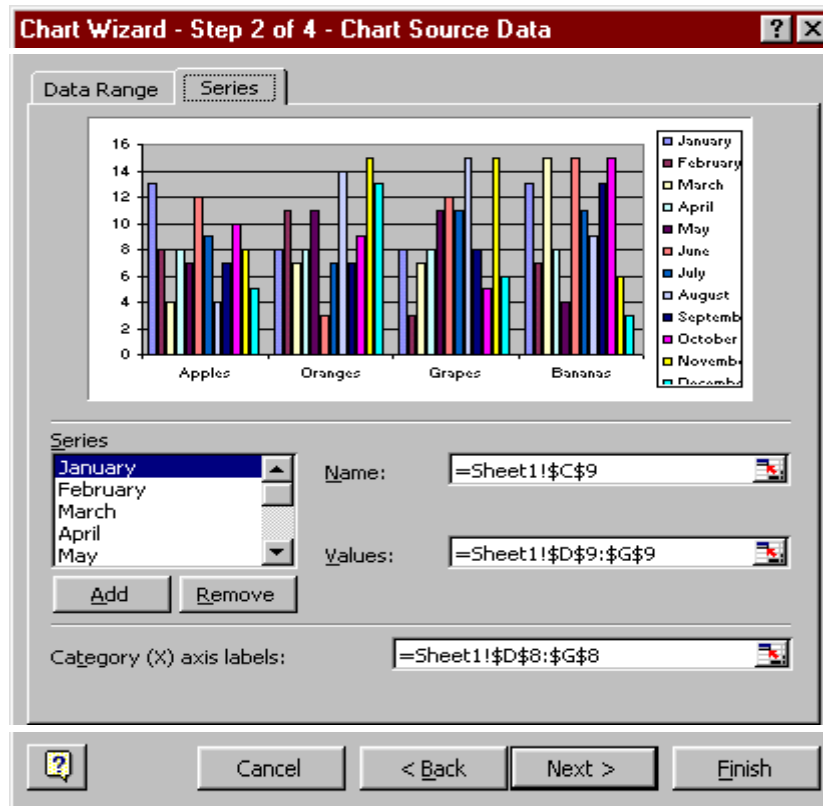
If you click in the Data range box, or click on the Range selection button , the dialog box shrinks to the size of just the data range box, obscuring less of the worksheet, so you are able to adjust this range with your mouse.



When the Source Data dialog box is activated, a “marching ants” border, shown in unanimated form below, highlights the source data range.

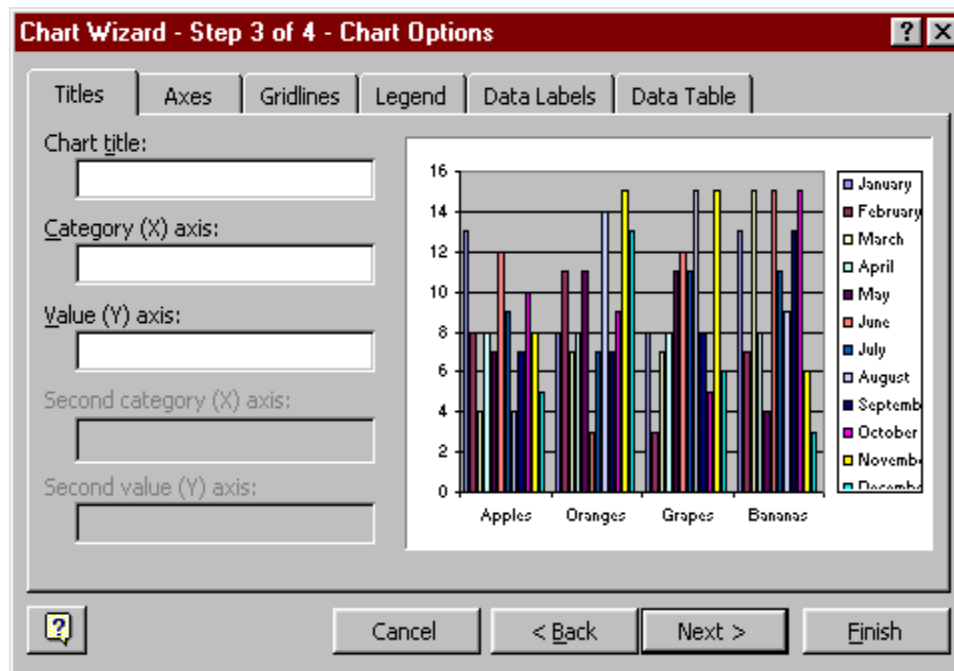
	Apples	Oranges	Grapes	Bananas
January	13	8	8	13
February	8	11	3	7
March	4	7	7	15
April	8	8	8	8
May	7	11	11	4
June	12	3	12	15
July	9	7	11	11
August	4	14	15	9
September	7	7	8	13
October	10	9	5	15
November	8	15	15	6
December	5	13	6	3

Clicking on the Series tab gives you the option to add or remove some of the data series in the selected source data range. You can change the Name, Values (Y axis), and Category labels (X axis) of each series, by editing the contents of the respective box, or by clicking on the Range selection button , then dragging with the mouse.



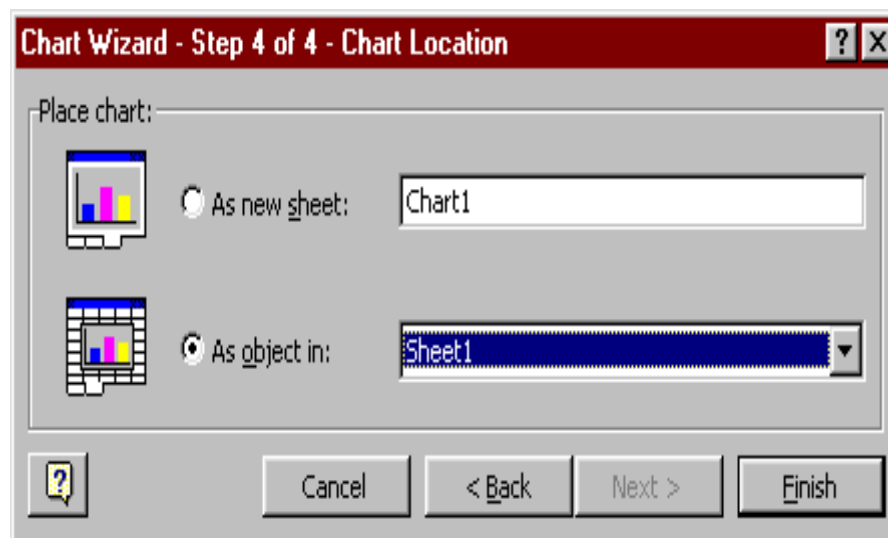
Step 3 - Chart Options

This step allows you to provide a title for the chart and for the chart axes, to select which and what kind of axes to put in your chart, and whether to show other options, such as gridlines, a legend, and data labels.



Step 4 - Chart Location

Here you tell Excel where to place the chart. You can put the chart onto its own sheet with a name that you enter in the box (where it says "Chart1") or you can embed the chart as a chart object in one of the existing worksheets (or chart sheets), which you select from the drop down list.



11.13. Summary:

In this lesson you learnt the basic features of Spread Sheet. Excel is used for performing accounting problems. In this chapter we introduced the concept of inserting, modifying and deleting rows/columns. We have learnt the procedure of changing the size of rows/columns and changing the data in a cell. Excel is having different types of references like Relative, Absolute and Named references. The data can be represented in many ways; using the different types of charts we can represent the data in effective manner.

11.14. Technical Terms:

Active Cell: The active cell has a dark border around it to indicate your position in the worksheet.

Relative Reference: Relative cell references are references to cells relative to the position of the formula.

Absolute Reference: Absolute references, which are cell references that always refer to cells in a specific location.

Formula Bar: Use the formula bar to enter and edit worksheet data.

11.15. Model Questions:

1. Explain the features o MS-Excel
2. Explain different references in MS-Excel
3. Explain the Chart feature in MS-Excel.
4. What is Auto Fill ? Explain Custom Fill List feature MS-Excel.

11.16. References:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, ' The ABC'S Of Microsoft Office. ' BPB Publications, New Delhi

MS OFFICE 2000 Complete, BPB Publications, New Delhi

C.V.P.R.PRASAD, M.C.A.,
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson –12

MICROSOFT EXCEL

12.0 Objectives:

This lesson introduces the basic features of Microsoft Excel, Pictures and Auto shapes, Worksheet Formatting, Worksheet Printing, Worksheet as Database and Macro in Excel.

Structure of the Lesson:

- 12.1. Pictures and Auto shapes
- 12.2. Worksheet Formatting
- 12.3. Functions
- 12.4. Worksheet as Data base
- 12.5. Worksheet Printing
- 12.6. Macro
- 12.7. Summary
- 12.8. Technical Terms
- 12.9. Model Questions
- 12.10. References

12.1. Pictures and Auto shapes:

Pictures

Excel has a feature of inserting pre-defined pictures or clipart into the worksheets. All these features are available under Insert Menu of the menu bar.

Inserting Clipart

Select the worksheet area in which you want to insert clipart. Select insert menu from the menu bar, click picture, and clipart. Now we see lot of pictures in a dialog box. Select Clipart tab, category, select an image and click insert. Then automatically the selected image will be displayed

in the worksheet and then figure will have small squares around the surface. These are called handles and are used to increase or decrease the size of the image.

Inserting picture form file

We can also insert pictures form a file. For this select insert menu from the menu bar and click picture from file. Then we see a dialog box in that select the required folder in which the file is available and the name of the file and click on the Insert button. Then the pictures from the file will be inserted into the worksheet.

Inserting Auto shapes

An auto shape can be inserted into a worksheet. For this select Insert menu from the menu bar and click picture and in it Auto shape. Then different shapes will appear in the dialog box. From that select the required auto shape. Now place the mouse pointer in the sheet where you want to draw the auto shape. The mouse pointer looks like a + sign. Then press the left mouse button and drag the mouse diagonally or towards right so that the selected auto shape will appear in the worksheet. After reaching the required auto shape then release the left mouse button. The size of the auto shape can be increased or decreased and also you can rotate the auto shape by an angle using the auto shape tool bar.

12.2. Worksheet Formatting:

Alignment of cells

Generally in Excel by default all the text will be aligned left and numbers to the right. But we can align text or number in the desired fashion. Select Format menu from the menu bar, Cells, then activate the Alignment Tab. There are two types of cell alignments they are Horizontal and Vertical alignments.

In the case of Horizontal alignment we have default option left (contents are aligned left), right (contents are aligned right), center (contents are aligned center), fill (contents are duplicated so that they fill the cells), justify (aligned left and right) options available.

In the case of vertical alignment we have Top (contents are aligned top), center (contents are aligned center), bottom (contents are aligned bottom), justify (contents are aligned top and bottom) options are available.

Excel allows Text Wrap option. When we type text, if the text overflows then adjacent cells will be used. But in the text wrap option there will not be any overflow instead the cell width will be increased to accommodate the text.

Number Formatting

Entering and formatting numbers

Numbers are referred as constant values or values. In addition to the numbers 0 to 9, you can enter the following symbols. That are

+ - () , . \$ %

It treats commas and other dolor signs correctly, and it accepts numbers entered in scientific notation (2.5E+2)

Different Number Formats

- General
- Number
- Currency
- Accounting
- Date
- Time
- Percentage
- Fraction
- Scientific
- Text
- Special
- Custom

When we enter any number it will take the number in general format. We can change the type of cell after entering the numbers in a cell.

For example If you activate a cell in the General format, then type a number with a dolor sign (\$), the cell's format will change from general to currency format.

When a number is too big to be properly displayed in its cell, excel after displays a series of pound signs (#####) instead of the number. Some times excel switch to scientific notation to accommodate a large number.

Entering Dates and Times

You can enter dates and times by typing item in most commonly accepted American formats.

11/7/97

11-Nov-97

Nov 7, 1997

The excel stores dates and times as serial numbers using January 1, 1900 at the stating date that in one (1). The serial number 2 stand January 2, 1990. You can change the date format to number, and number to date. If we enter 8/93, the system automatically displays accepts as Aug-93.

If we enter numbers starting with ' (single quote), the entered data will be treated as text.

Fonts for Cells

To change the, font size, font style, Bold, Italic and Underline, select the cells in which we want to change the font. Select Format menu from the menu bar, Cells, then activate the Font Tab. Now apply font style, size, bold, italic etc., and click OK. Now the selected cells will be formatted accordingly.

Auto Format

Excel has some pre-defined formatting schemes, and those can be utilized directly instead of formatting worksheet.

Select the cells, which are to be auto formatted and then select Format from the menu bar and click Auto Format. Then we get auto format dialog box on the screen. Here click the required format style, and click O.K. Then the selected cells will be formatted according to the selected pre-defined auto format.

12.3. Functions:

The various MS-Excel functions are classified as:

1. Statistical function:

Function	Purpose
SUM()	It computes the sum of a particular defined range or range names. Ex: sum(a1:a4)
Average()	It computes the average of a particular defined range or range numbers. Ex:- Average(a1:a4)
Count()	It counts the number of non blank cells in a particular defined range or range names. Ex:- count(a1:a4)
Max()	It will display the maximum value from a particular defined range or range names. Ex:- max(a1:a4)
Min()	It will display the minimum value from a particular defined range or range names. Ex:- min(a1:a4)

2. Mathematical function:

Function	Purpose
ABS()	It will calculate the absolute value of a number. Ex:- abs(-9) will display as 9.
Int()	It will round off the number down to its nearest integer. Ex:- int(19.34) will display as 19
Mod()	It will calculate the remainder after integer division. Ex:-mod(5,2) will display 1
Round()	It will round off the number down to a specified number of decimal places. Ex:- round(a1,2) will display 19.34 if cell a1 contains 19.345678
Sqrt()	It will calculate the square root of a positive number. Ex:- sqrt(9) will display 3.

3. Trigonometric functions:

These functions are used for trigonometric calculations on a list of values. Some commonly functions like SIN, COS, TAN, ACOS, ATAN, ACOSH, ATANH and DEGREES etc.

Syntax:- function (cell address/value)

4. Financial functions:

These functions are used to compute loans, depreciation, cash flow etc., Some commonly used financial functions are RATE, FV, PMT etc.,

Rate(): It calculates the rate of interest per period. The syntax is RATE(nper, pmt, pv, fv, type, guess). Here nper is the total payment period, pmt is payment made per period, pv is present value of the total amount, fv is the future value, type is the number 0 or 1 depending upon whether the payment is to be made at the end of the period or at the beginning respectively. Guess is the guess value for the rate of interest.

For example, amount of loan = Rs.50,000 to be paid in 24 monthly installments of 3,000 each. Here nper = 24, pmt=-3,000, pv=50,000 fv=0.The default guess is 10%. Now RATE(24,-3000,50000,0,0,10) will display 1% as rate of interest.

PMT(): The purpose of PMT() function is to compute the equal periodic payments required to pay off a loan. The general syntax of PMT() function is PMT(rate, nper, pv, fv, type)

Here fv and type are optional. Suppose if you want to take a loan of Rs.2,00,000/-. The annual interest rate is 14% and has to be paid in 15 years in equal monthly installments.

Here rate is 14%/12, nper is 180 (15 * 12) and pv is 2,00,000. Here fv is zero by putting all these in the formula

Pmt(14/12,180,200000).

12.4. Worksheets as Data Bases:

Excel is a spreadsheet application but can also be used as a database. There are two limits for using Excel as a database. The first one is that only one person at a time can enter data into the database.

The second limit is a matter of quantity of data. Each sheet in an Excel workbook comprises 65,000 rows and 256 columns. An Excel workbook can comprise up to 256 sheets. We are talking about 4,292,608,000 cells.

Excel needs to recognize your set of data as a database or you will not have access to any of the database functionalities from the "Data" menu item (Sort, Filter, Form, Subtotal).

The DATABASE is a set of columns and rows in your spreadsheets without empty rows and empty columns. Many cells can be empty but never an entire row or an entire column.

The database must have a single title row. Use "Text Wrap" in Format/Cells/Alignment to write more than one line of text in one cell.

The DATABASE must be surrounded by empty rows and columns (row "1" and column "A" of the worksheet are considered preceded by an empty row or column).

So anybody can develop a database in Excel. You don't need to develop complex forms to enter data, you don't need to develop queries you just use SUMPRODUCT, INDEX/MATCH and SUBTOTAL formulas to develop reports by yourself when you need them and that have the layout that you need.

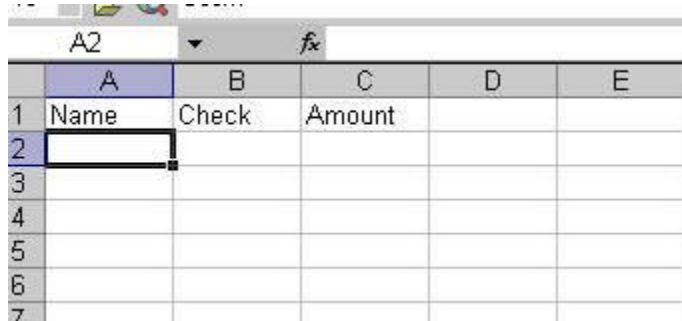
Excel is the most user friendly analyzing and reporting application so they don't need to train a few analysts on other reporting applications creating a bottleneck at the report development level. Adopting Excel also allows all the analyst and decision makers to develop significant analysis and reports improving the bottom line of the whole corporation.

Using Data Forms

A *data form* is used to allow easy manipulation of information in an Excel data list. While a list is small--for instance, when it fits on one screen--it is easier to enter or change information directly. When you start getting a larger number of records, then you may find using a data form to be easier. A data form is a dialog box that displays one complete record from your list at a time. Excel considers a record to be a single row in your data list, so a data form basically extracts the information from a row, uses the field labels from the first row of the list, and displays the information so you can understand it easier. To utilize a data form, follow these two simple steps:

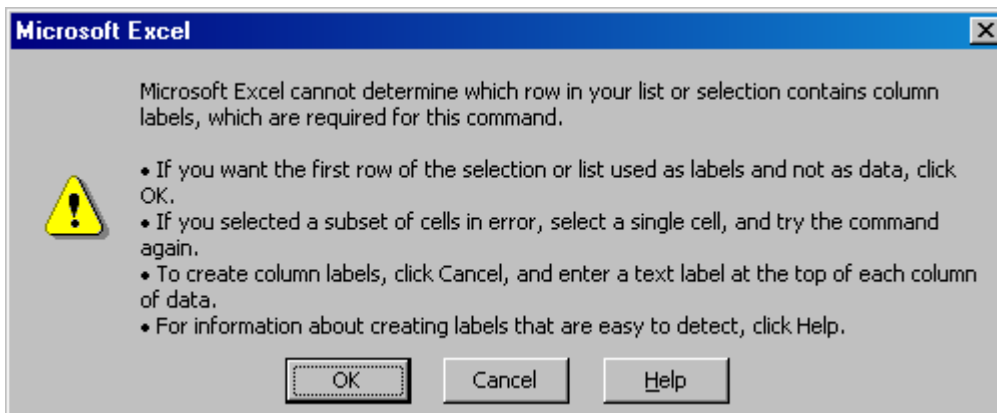
Creating Data Forms in Excel

When you have to input a data into a worksheet, using an Excel generated data form may make the job easier. Your worksheet should contain column titles as in the example below:



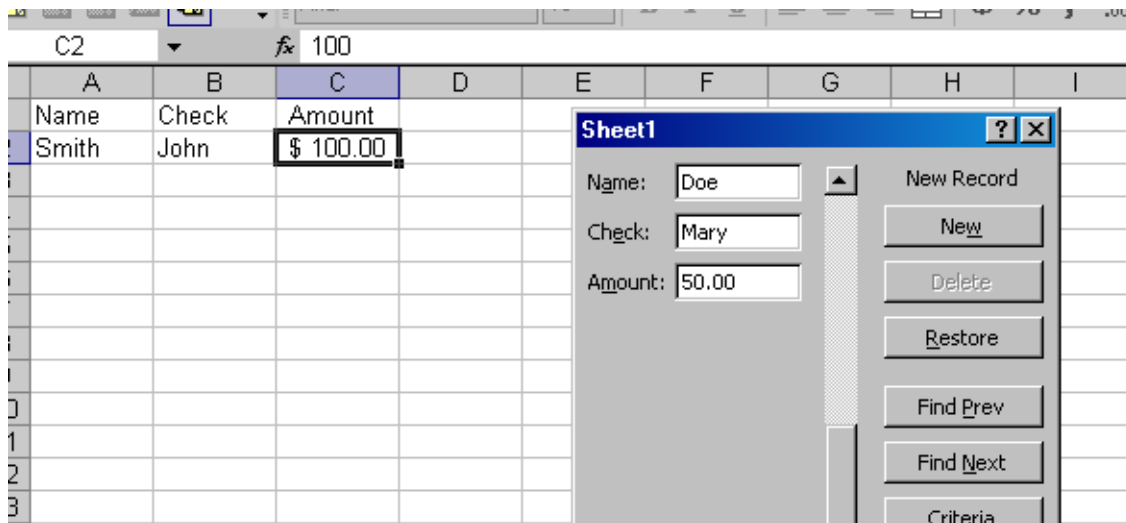
	A	B	C	D	E
1	Name	Check	Amount		
2					
3					
4					
5					
6					
7					

Click in the first cell in the row directly below your titles, then select Data, Form. If you receive the message below, click OK.



Your data form is created and ready for data entry.

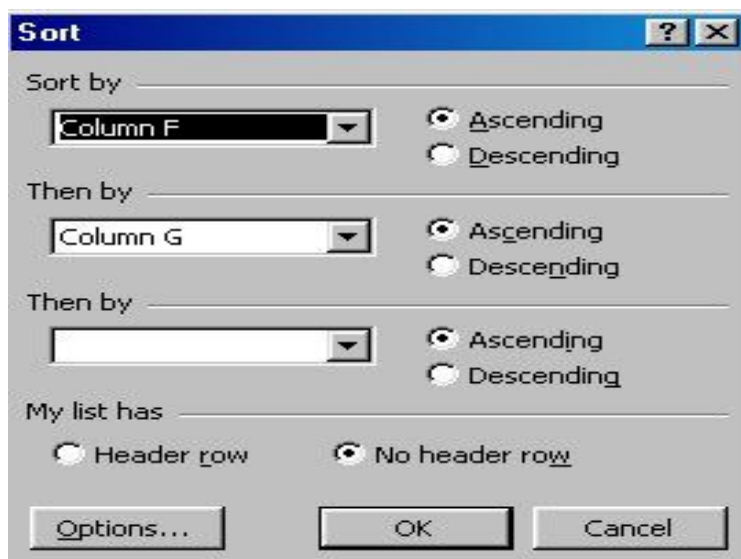
Type your data, tabbing between fields and press Enter or click New after each record to add it to the worksheet.



When you are done entering records, click Close.

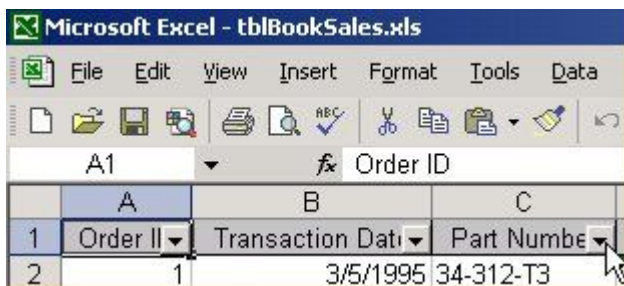
Sorting Data:

1. Select the columns that you want to sort.
2. Click Data>Sort.
3. In the sort window choose the column or columns you want to sort by.
4. If you have a heading row make sure the Heading row option is selected.
5. Click OK.
6. The data will now be sorted.



Filter Data:

1. Click **Data>Filter>AutoFilter**.
2. The first row of the spreadsheet becomes the filtering row.
3. To filter for specific data, click on the down arrow next to the head of the column you want to filter by.
4. Then select the criteria you want to filter for.
5. You may filter by multiple columns at once by repeating steps 3-4 for each column you want to filter by.



12.5. Worksheet Printing:

Page Breaks

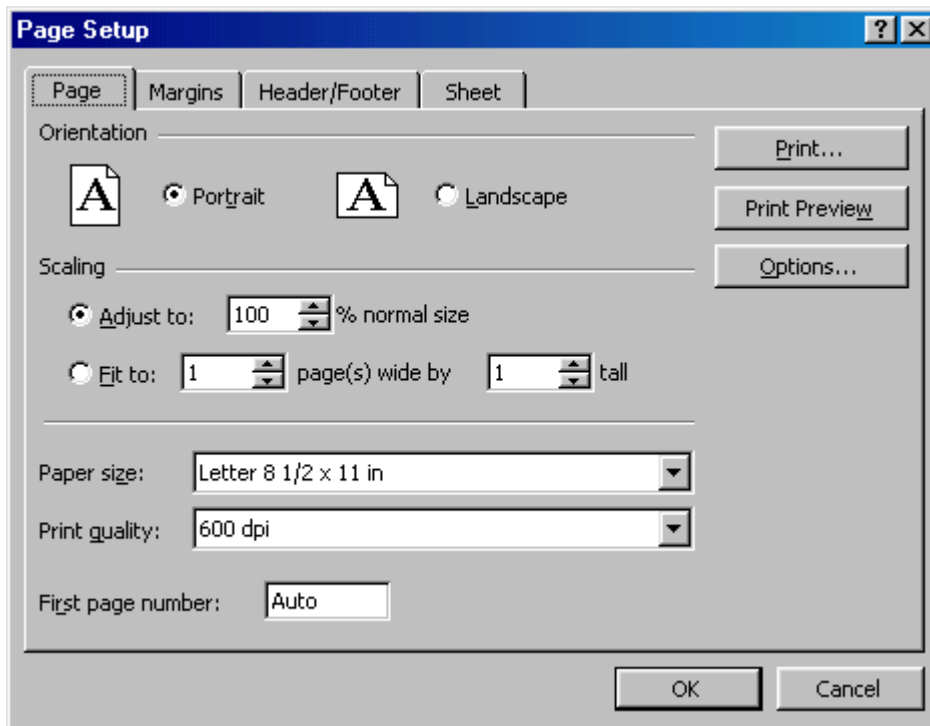
To set page breaks within the worksheet, select the row you want to appear just below the page break by clicking the row's label. Then choose **Insert|Page Break** from the menu bar. You may need to click the double down arrow at the bottom of the menu list to view this option.

Page Setup

Select **File|Page Setup** from the menu bar to format the page, set margins, and add headers and footers.

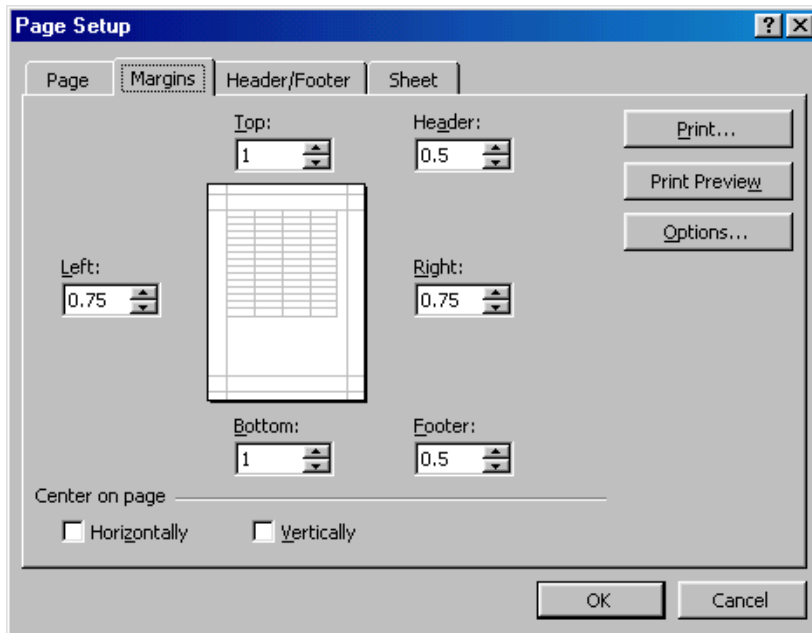
Page

Select the **Orientation** under the **Page** tab in the Page Setup window to make the page Landscape or Portrait. The size of the worksheet on the page can also be formatting under **Scaling**. To force a worksheet to print only one page wide so all the columns appear on the same page, select **Fit to 1 page(s) wide**.



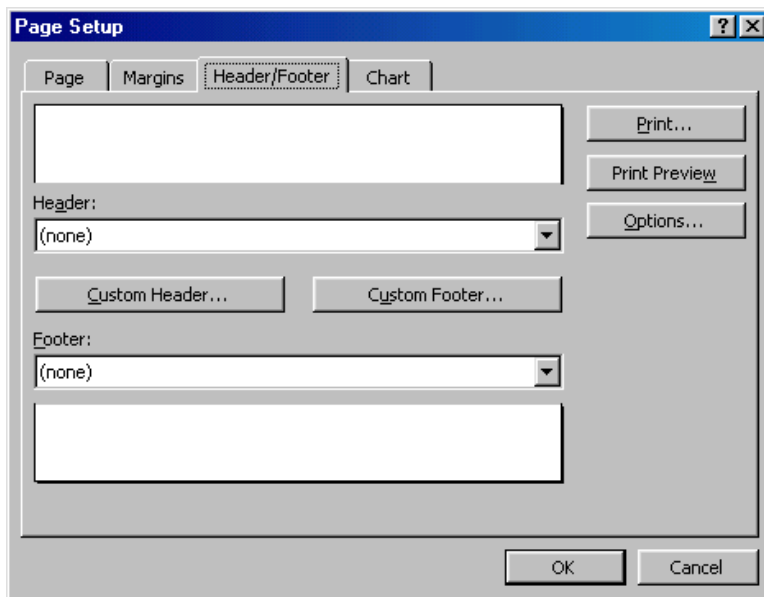
- **Margins**

Change the top, bottom, left, and right margins under the **Margins** tab. Enter values in the header and footer fields to indicate how far from the edge of the page this text should appear. Check the boxes for centering horizontally or vertically on the page.

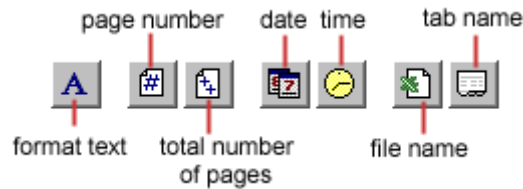


- **Header/Footer**

Add preset headers and footers to the page by clicking the drop-down menus under the Header/Footer tab.



To modify a preset header or footer, or to make your own, click the **Custom Header** and **Custom Footer** buttons. A new window will open allowing you to enter text in the left, center, or right on the page.



Format Text - Click this button after highlighting the text to change the font, size, and style.

Page Number - Insert the page number of each page.

Total Number of Pages - Use this feature along with the page number to create strings such as "page 1 of 15".

Date - Add the current date.

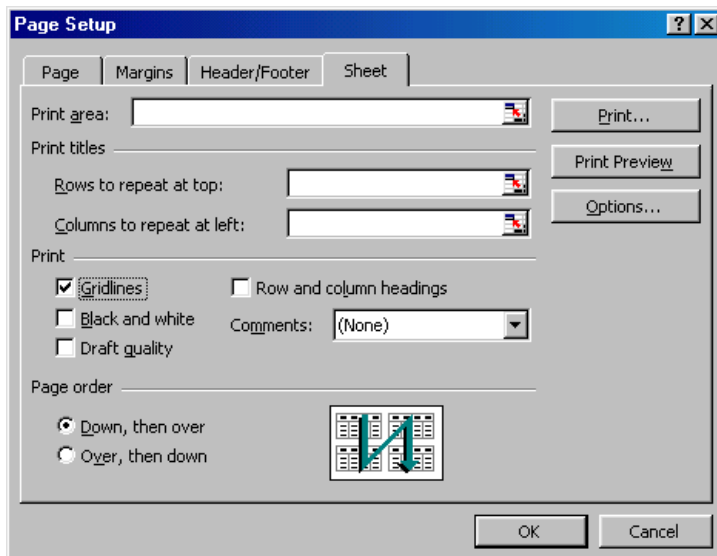
Time - Add the current time.

File Name - Add the name of the workbook file.

Tab Name - Add the name of the worksheet's tab.

- **Sheet**

Check **Gridlines** if you want the gridlines dividing the cells to be printed on the page. If the worksheet is several pages long and only the first page includes titles for the columns, select **Rows to repeat at top** to choose a title row that will be printed at the top of each page.

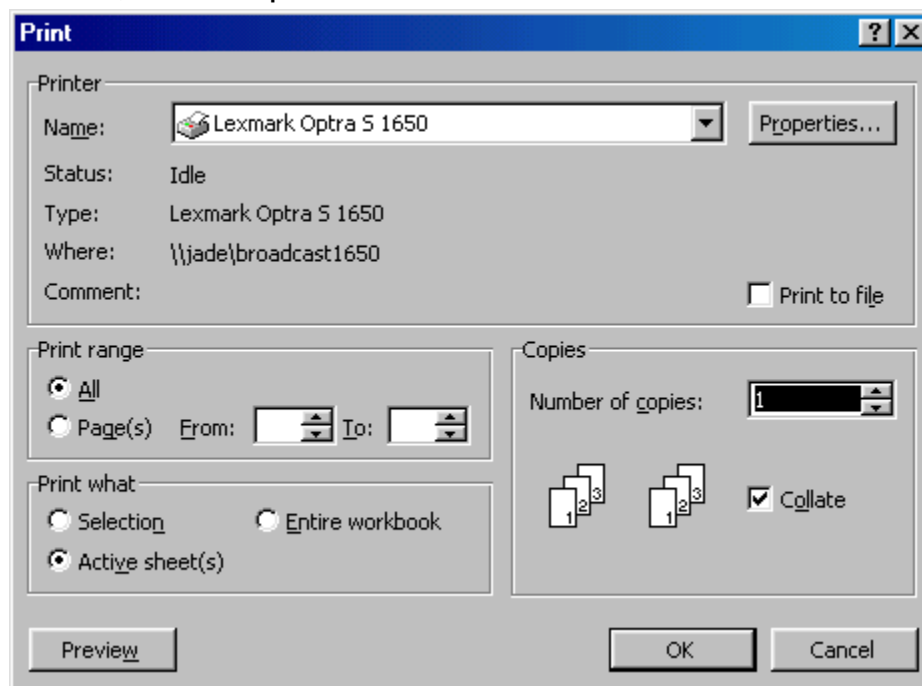


Print Preview

Select **File|Print Preview** from the menu bar to view how the worksheet will print. Click the **Next** and **Previous** buttons at the top of the window to display the pages and click the **Zoom** button to view the pages closer. Make page layout modifications needed by clicking the **Page Setup** button. Click **Close** to return to the worksheet or **Print** to continue printing.

Print

To print the worksheet, select **File|Print** from the menu bar.



- **Print Range** - Select either all pages or a range of pages to print.
- **Print What** - Select selection of cells highlighted on the worksheet, the active worksheet, or all the worksheets in the entire workbook.
- **Copies** - Choose the number of copies that should be printed. Check the **Collate** box if the pages should remain in order.

Click **OK** to print.

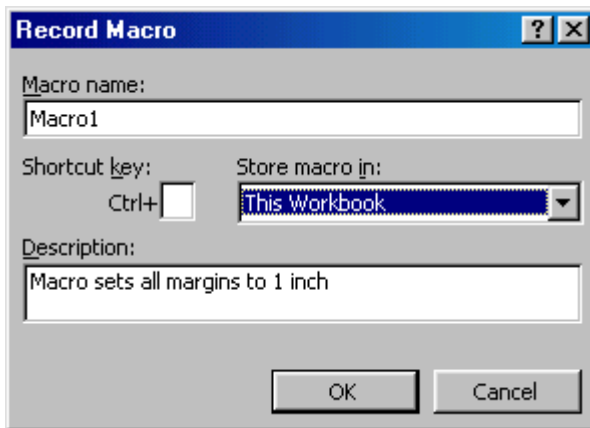
12.6. Macro:

Macros can speed up any common editing sequence you may execute in an Excel spreadsheet. In Macro two options are there. They are Recording a Macro and Running a Macro.

Recording A Macro

In this example we will make a simple macro that will set all the margins on the page to one inch.

- Click **Tools|Macro|Record New Macro** from the menu bar.



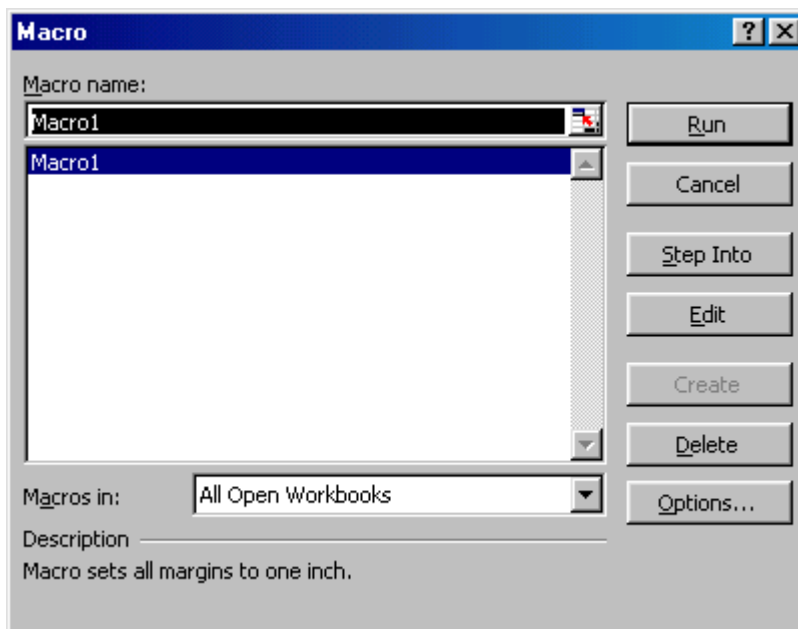
- Name the macro in the **Macro name** field. The name cannot contain spaces and must not begin with a number.
- If you would like to assign a shortcut key to the macro for easy use, enter the letter under **Shortcut key**. Enter a lower case letter to make a CTRL+number shortcut and enter an upper case letter to assign a CTRL+SHIFT+number shortcut key. If you select a shortcut key that Excel already uses, your macro will overwrite that function.
- Select an option from the **Store macro in** drop-down menu.
- Enter a description of the macro in the **Description** field. This is for your reference only so you remember what the macro does.
- Click **OK** when you are ready to start recording.
- Select options from the drop down menus and Excel will record the options you choose from the dialog boxes, such as changing the margins on the Page Setup window. Select **File|Page Setup** and change all the margins to 1". Press **OK**. Replace this step with whatever commands you want your macro to execute. Select only options that modify the worksheet. Toggle actions such as **View|Toolbars** that have no effect on the worksheet will not be recorded.



- Click the **Stop** button the recording toolbar. The macro is now saved.

Running A Macro

- To run a macro you have created, select **Tools|Macro|Macros** from the menu bar.
- From the **Macros** window, highlight the **Macro name** in the list and click **Run**.



- If the macro is long and you want to stop it while it is running, press **BREAK** (hold **CTRL** and press **PAUSE**).

12.7. Summary:

In this lesson we have learnt how to insert graphics in Excel. You have learnt the usage of functions. Functions are differentiated into different types. We can use the functions with respect to our application. It is the best way to see the output in print preview fashion before we are going for final print. Worksheet can also be used as a Database. You have learnt the concept of using Data Forms. By using the Data Form we can enter the data and treat it as a record. We can sort the data and can filter the data with respect to our criteria.

12.8. Technical Terms:

Data Form: A *data form* is used to allow easy manipulation of information in an Excel data list.

Page Orientation: Page Orientation is used to set the page Landscape or Portrait.

12.9. Model Questions:

1. Explain the usage of functions of Excel?
2. Explain the procedure to use the Data Forms in Excel?
3. Explain the concept of formatting data in excel?
4. How you can use the worksheet as a Database? Explain.

12.10. References:

RON MANSFIELD, 'Working with MS OFFICE', Tata Mc Graw Hill – 2000 Edition

GUY HART DAVIS, ' The ABC'S Of Microsoft Office. ' BPB Publications, New Delhi

MS OFFICE 2000 Complete, BPB Publications, New Delhi

C.V.P.R.PRASAD, M.C.A.,
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson – 13

INTRODUCTION TO 'C'

13.0 Objectives:

The main objectives of this lesson are:

- To familiarize the students with the history and development of C language.
- To clarify certain basic concepts in C language.
- To know the different elements of C like character set, keywords etc.,
- To classify different C program entities like identifiers, constants, statements and expressions.
- To know different types of data types available in C language.
- To understand the working and usage of input/output statements in C language.

Structure of the Lesson:

- 13.1. Introduction
- 13.2. History and Development of C
- 13.3. Features of C
- 13.4. Types of C compilers
- 13.5. Structure of a C program
- 13.6. C character set
- 13.7. Escape sequences
- 13.8. C reserved words
- 13.9. Identifiers
- 13.10. Data types
- 13.11. Constant
- 13.12. Variables
- 13.13. Expression
- 13.14. Statements
- 13.15. Symbolic constants
- 13.16. Operators
- 13.17. scanf()
- 13.18. printf()
- 13.19. Summary
- 13.20. Technical terms
- 13.21. Model questions
- 13.22. References

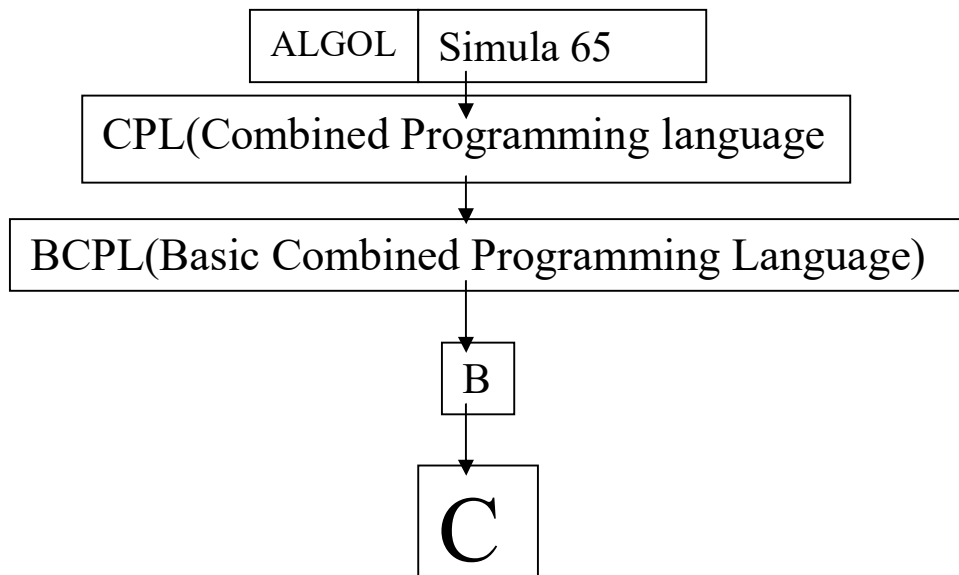
13.1.Introduction:

The step-by-step process of solving a given problem is called an algorithm. Using the algorithm select a programming language of your choice, based on the type of problem. Convert the algorithm into a program following the syntax of that language. One of such programming languages is our C.

13.2. History and Development of C Language:

C is an offspring of the Basic Combined Programming Language (BCPL) and B developed in 1960 at Cambridge University. The language B was modified by Dennis Ritchie and implemented at Bell Laboratories in 1972. This modification was named as C.

'C' was then used to implement the UNIX operating system. C is a general-purpose language, which has been closely associated with the UNIX operating system.



13. 3.Features of C language:

C is a beautiful language and provides many built-in functions that help in problem solving. We list out certain common features of C that have been quite useful.

- C is a powerful, flexible language that provides fast program execution and imposes few constraints on the programmer.
- C's power and fast program execution come from its ability to access low level commands, similar to assembly language, but with high level syntax.
- Another strong point of C is its use of modularity. Sections of code can be stored in libraries for re-use in future programs. This concept of modularity also helps with C's portability and execution speed.
- C is a block-structured language with fundamental flow-control construction.
- C includes certain low-level features that are normally available in assembly or machine languages.
- Programs written in C compile into small object programs that execute efficiently.
- C is a largely machine independent language and programs written in C can be easily ported from one computer to another.

- C compilers are widely available in most personal computers minicomputers and mainframes.

13.4.Types of C compilers:

The following are the some of the compilers for C language:

- Turbo C
- Quick C
- Microsoft C

13.5.Structure of a C program:

Every C program consists of one or more functions, one of which must be called main. The program will always begin by executing the main function. Additional function definitions may precede or follow main(). Hence we say that C is a function dependent language.

Structure of a C program:

```
Documentation section
Include libraries - Linkage section - Preprocessor Directives
Declare global variables
Declare constants
Define macros
Function Prototype Declaration Section
main()
{
Local variable declarations
-----//
Main body of the program
-----//
}
function-1()
{
-----
}
function-2()
{
-----
}
.....
Function-n()
{
-----
-----
}
```

Here

1. Comments are non-executable statements meant for remarks regarding program. Comment lines begin with a /* and end with a */.
2. 'C' is very case sensitive. All 'C' statements should be in lower case except user-defined expressions.
3. Every statement in 'C' must be terminated by a semicolon except the conditional statements like if, if-else, switch, for, while and compile directive statements like #include, #define.
4. Every statement must be included in { and } of main().
5. Every program must include at least one function called main()

13.6.C Character Set:

Character set means the characters and symbols that can be used in a C program. They are grouped to form the commands, expressions and other tokens for C language. Character set is the combination of characters, digits, special characters and blank spaces. C language uses

Alphabets:	upper case letters A to Z Lower case letters a to z
Digits:	0 to 9
Special Characters:	{,}, #, : ,;, &, ? Etc.,

13.7.Escape Sequences:

A special kind of a character that allows the user to customize the output of a program is referred as escape sequence.

\b	back space
\n	new line
\t	horizontal tab
\a	bell
\\	backslash
%%	percentage
\v	vertical tab
\0	NULL character
\f	form feed

Ex: **Program to print "Hello Ramu" on the screen.**

```
#include <stdio.h>
#include <conio.h>
main()
{
    printf("\n\n\n\n\n\n\t\t\t Hello Ramu");
```

```

printf("\n\a This is JKC College");
printf("\n\t Dept.Of Computer Science);
printf("\n Guntur.");
getch();
}

```

13.8.Keywords:

There are certain reserved words, called keywords that have standard, predefined meanings in C. These keywords are to be used for their intended purpose only in a C program. There are 32 words defined as keywords in C. They are always written in lower case. A complete list follows:

auto	double	int	struct
Break	else	long	switch
case	enum	register	typedef
char	extern	return	union
Const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

In addition to this list of keywords, your compiler may define a few more. If it does, they will be listed in the documentation that comes along with your compiler.

13.9.Identifiers:

Identifiers are names given to various program elements, such as variables, functions and arrays. Identifiers consist of letters and digits, in any order, except that the first character must be a letter. Both uppercase and lowercase are permitted. Upper and lowercase letters are not interchangeable, i.e., uppercase letter is not equivalent to the corresponding lowercase letter and vice versa. The under score character (_) can also be included. An underscore is often used in the middle of an identifier.

Valid Identifiers

A, B, Stno, tax_rate etc.,

Invalid Identifiers

4th "x"

13.10.Data Types:

C supports several different types of data, each of which may be represented differently within the computer's memory. A data type is a C token that tells about the type of data being assigned to an identifier.

C supports a rich set of data types. The category follows:

- Primary data types or Primitive data types
- Secondary data types

While using the above data types, one should intimate the compiler by providing corresponding format specifier or control string.

Primary data types:

Short int
int
long int
float
double
long double

Secondary data types:

Arrays
Functions
Pointers
Structures
Unions
Enumeration

The % format specifiers:

The % specifies that you can use in ANSI C are:

Format	Data Type
%d (%i)	Int
%e (%E)	float or double
%f	float or double
%o	Octal
%p	Pointer
%s	String
%u	Unsigned data representation
%x (%X)	Hexa-decimal data representation
%c	Character (single)
%ld	Long int
%lf	Long double

The data size of each data type follows:

Data type	Data size in bytes	Range
Short int	1(8 Bits)	-128 to 127
Unsigned short Int	1(8 bits)	0 to 255
Int	2(16 bits)	-32768 to 32767
Unsigned int	2(16 bits)	0 to 65535
Long int	4(32 bits)	-2,147,483,648 to 2,147,483,647

Double	8(48 bits)	1.7E-308 to 1.7E+308
Long double	10(80 bits)	3.4E-4932to 1.1E+4932
Char	1(8 bits)	-128 to 127

13.11. Constants:

A constant is an identifier whose value can never be changed during the program execution. C has four basic types of constants. They are

- 1.Integer constant
- 2.Floating-point constants
- 3.Character constant
- 4.String constant

Integer constant:

0 1 743 5280

Floating constants:-

0. 1. 0.2 825.602 2E-8 0.06E-3

3*10⁵ can be represented as follows:

300000. 3E5 3E+5 3.0E+5 .3E+6 30E4

5.026*10⁻¹⁷ can be represented as follows:

5.026E-17
.5026E-16

Character constants:-

'A', 'X', '3', '\$'

String Constants:

A string consists of any number of consecutive characters enclosed in double quotations.

Ex: "green", "jkc college"

13.12.Variable:

A variable is an identifier that is used to represent some specified type of information. Or a variable is an identifier whose value may or may not be changed during the program execution.

int a, b, c;

char d; a=3; b=4; d = 'a';

All variables must be declared before they can appear in executable statements.

int a, b, c; float root1, root2;
char flag;

13.13.Expression:

An expression represents a single data item, such as a number or a character. It may also consist of a combination of entities interconnected by one or more operators.

Ex: c=a+b
 x=y;
 x<=y
 x==y
 ++i;

13.14.Statements:

A statement causes the computer to carry out some action. There are three different classes of statements in C.

- Expression Statement
- Compound Statement
- Control Statement

An expression statement consists of an expression followed by a semicolon.

```
a=3;  
c=a+b;  
a++;  
printf("jkc college");
```

A compound statement consists of several individual statements enclosed within a pair of braces {}.

```
{  
  pi=3.1415;  
  area=pi*radius*radius;  
}
```

Control statements are used to create special program features, such as logical tests, loops and branches.

Ex: Program to add two numbers 10 and 20.

```
#include <stdio.h>  
#include <conio.h>  
main()  
{  
    int a, b, sum;  
    clrscr();  
    a = 10;  
    b = 20;
```

```

    sum = a + b;
    printf("\n\t Sum of %d and %d is %d", a, b, sum);
}

```

13.15. Symbolic constants:

A symbolic constant is a name that substitutes for a sequence of characters. The characters may represent a numeric constant, a character constant or a string constant.

```
# define name text
```

Where name represents a symbolic name, typically written in upper-case letters, and text represents the sequence of characters associated with the symbolic name. Note that text does not end with a semicolon.

```

# define TAXRATE 0.25
# define PI 3.1415
# define TRUE 1
# define FALSE 0

```

Notice that the symbolic names are written in uppercase, to distinguish them from ordinary C identifiers.

13.16. Operators:

An operator performs an operation over one or more operands. C supports a rich set of operators. The following are the some of the examples.

Unary Operator:

An unary operator performs operation on one operand. The unary operators are: ++, --

Ex:

```

→ ++a or a++ is equivalent to a=a+1
→ --a or a-- is equivalent to a=a-1.
Here ++a - Pre-incrementation
      a++ - Post-incrementation
      - -a - Pre-decrementation
      a- - - Post-decrementation

```

- In Pre-incrementation the value is incremented first, then assigned.
- In Post-incrementation, the value is assigned first, then incremented.
- In Pre-decrementation, the value is decremented first, then assigned.
- In Post-decrementation, the value is assigned first, then decremented.

For example:

```

int i = 5;
printf ("%d\n",i);    →    5
printf ("%d\n",i++); →    5
printf ("%d\n",i);    →    6
printf ("%d\n",++i);  →    7
printf ("%d\n",i);    →    7
printf ("%d\n",i--);  →    7
printf ("%d\n",i);    →    6
printf ("%d\n",--i);  →    5
printf ("%d\n",i);    →    5

```

Binary Operators:

A binary operator is an operator that performs operations on two operands or values. There are many binary operators in C. Some of them are: +, -, *, /, %, +=, -=, *=, /= etc.,

Arithmetic operators are the regular operators that can be used to perform basic arithmetic operations like addition subtraction, multiplication and division in a program in order to solve a problem.

Operators	Purpose
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Reminder

Suppose a and b are integer variables such as a=10 b=3

Expression	Value
a+b	13
a-b	7
a*b	30
a/b	3
a%b	1

v1 v2 are floating points variables whose values are 12.5 and 2.0

Expression	Value
v1+v2	14.5
v1-v2	10.5
v1*v2	25.0
v1/v2	6.25

The value of an expression can be converted to a different data type if desired. To do so the expression must be preceded by the name of the desired data type, enclosed in parentheses.

(Data type) expression

```
int a=10;
int b=3;
a/b becomes 3
(float)a/(float)b becomes 3.33
int 10+15.5 becomes 25
```

Relational Operators:

Operators	Meaning
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal to
!=	not equal to

Logical Operators:

Operator	Meaning
&&	and
	or
!	not

Assignment Operators:

In C there are different assignment operators available. They are: =, +=, -=, *=, /=, %=.

Syntax: identifier = expression;

Example: a=3;
x=y;
sum=a+b;
a += 4; // a = a+4;
a -= 4; // a = a- 4;
a *= 4; // a = a4;
a /= 4; // a = a/4;
a %= 4; // a = a%4;

13.17.Scanf function:

In order to provide input to any program one has to use a standard input statement in C. Another way to input data into the computer is through a standard input device using C library function scanf(). This function can be used to enter any combination of numerical values, single characters and strings.

```
scanf(control-string, arg1,arg2,... argn);
```

Where control string refers to a string containing certain required formatting information and arg1, arg2...argn are arguments that represent the individual input data items.

```
%c  single character
%d  decimal integer
%e  floating point value
%f  floating point value
%s  string
%x  hexa decimal
%o  octal
```

Each variable name must be preceded by an ampersand (&). The arguments are actually pointers that indicate where the data items are stored in the computers memory.

Example:

```
int a;
float b;
char c;
scanf("%d%f%c",&a,&b,&c);
char name[20];
scanf("%s", name);
```

13.18.printf Function:

A standard output statement with which the user is able to display or view the result or output of a program is nothing but a printf() statement. printf() is a C library function. This function can be used to output any combination of numerical values, single characters, and strings.

```
printf( control-string,arg1,arg2,...);
```

Where control string refers to a string that contains formatting information, and arg1, arg2 are arguments that represent individual output data items.

```
%c  single character
%d  decimal integer
%e  floating point value
%f  floating point value
%s  string
%x  Hexa decimal
%o  Octal
```

However, both printf() and scanf() are the standard input and output statements, they can be referred as formatted I/O statements or standard I/O.

Ex:Program to accept integer, float and character values and display them back.

```
main()
{
```

```
int a;
float f;
char ch;
clrscr();
printf("\n Enter an integer          : ");
scanf("%d", &i);
printf("\n Enter a floating number    : ");
scanf("%f", &f);
printf("\n Enter a character          : ");
scanf("%c", &ch);
printf("\n Integer value              : %d", i);
printf("\n Floating value             : %f", f);
printf("\n Character value            : %c", ch);
}
```

Program to add, subtract, multiply and divide two numbers

```
#include <stdio.h>
#include <conio.h>
main()
{
    int num1, num2, sum, diff, prod;
    float division;
    printf("Enter any two numbes");
    scanf("%d%d",&num1,&num2);
    sum  = num1 + num2;
    diff = num1 - num2;
    prod = num1 * num2;
    division = num1 / num2;
    printf("\n Sum is %d",sum);
    printf("\n Difference is %d. ",diff);
    printf("\n\n\t Product is %d. ",prod);
    printf("\n\n\t Division is %d. ",division);
}
```

Program to find area and perimeter of a rectangle given its length as 5 units and breadth as 10 units.

```
#include <stdio.h>
#include <conio.h>

main()
{
    int length, breadth;
    int area, perimeter;
    area = length * breadth;
```

```
    perimeter = 2 * (length + breadth);
    clrscr();
    printf("\n\n\t Length of Rectangle    :%d ",length);
    printf("\n\n\t Breadth of Rectangle    :%d ",breadth);
    printf("\n\n\t Area of Rectangle        :%d ",area);
    printf("\n\n\t Perimeter of Rectangle :%d ",perimeter);
}
```

13.19. Summary:

In this lesson you learned history and development of C language and basics required to write a C program: character set, keywords, data types, operators and input/output functions. You learnt that there are three basic data types in C: integer, float and character. You have learnt how to declare a variable, how to define a constant and how to use the arithmetic statements.

Input/output statements you have learnt in this lesson are scanf() and printf(). You have also learnt how to use these functions for input and output of different types of variable like integer, float, character, string etc.

13.20. Technical Terms:

Algorithm: A step-by-step procedure for solving a problem or accomplishing a task.

Structured programming: Is a design approach where complex systems are broken down into smaller, more manageable pieces.

ANSI[American National Standards Institute]: This organization produces many standards, among them the standard for the C programming language.

Escape Sequences: A special sequence of characters used for describing nonprinting characters, such as '\n' for new line.

Keyword: A keyword is a word that has special meaning. Keywords are reserved for specific purpose and not to be used as variable names.

Built-in Function: The language provides built-in functions that perform various numerical and string computations.

Variable: A name used to represent data that can be changed while the program or procedure is running.

Constant: A specific quantity that does not or cannot change or vary; the opposite of a variable.

Token: It is a smallest individual program element of a C program.

Preprocessor: It is a program, which processes the source code before it passes through the compiler.

13.21. Model questions:

1. What is C? Explain the history of C language?
2. Explain the features of C language?
3. Write about the structure of a C language program?
4. Explain various data types in C?
5. Explain different types of operators available in C?
6. Write about printf() and scanf() statements?

13.22. References:

Byron C. Gottfried “**Programming with C**”

Kelly & Pohl “**A book on C** “

Yeshavanth kanethkar “**Let us C**”

Ravi Chandran “**Programming in C**”

Yeshavanth kanethkar “**Working with C**”

Blaguruswamy “**ANSI C**”

- **G.RAMANJIAH.,M.Sc(Computers)**
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson -14

CONTROL STRUCTURES

14.0 Objectives:

The main objectives of this lesson are:

- To use control structures to write effective programs
- To understand various types of control structures
To develop C programs using decision making constructs **Objectives:**
-
- To develop C programs using iterative constructs
- To understand break, continue, goto and label statements

Structure of the Lesson:

- 14.1. Introduction
- 14.2. What are Control structures?
- 14.3. Conditional branching
 - 14.3.1. Simple if statement
 - 14.3.2. if else statement
 - 14.3.3. nested if statement
 - 14.3.4. switch statement
- 14.4. Looping structures
 - 14.4.1. while statement
 - 14.4.2. do while statement
 - 14.4.3. for statement
- 14.5. break and continue
- 14.6. goto and exit statements
- 14.7. Summary
- 14.8. Technical terms
- 14.9. Model questions
- 14.10 References

14.1.Introduction:

Every C program is a sequence of instructions, which are executed, in a sequential fashion. There are many situations in which these sequential executions need to be altered with a few instructions being executed only when a condition is satisfied. Sometimes it may be necessary to execute a group of instructions for a fixed number of times or until some condition is satisfied. In such

situations programmer may require to include control statements in programs. These control statements allow the programmer to alter the sequential order of execution.

14.2. Control structures:

In high-level programming languages, flow of program execution may be changed using certain control statements called control structures. A control structure is a **control flow statement** that allows you to alter the **sequential flow**.

Control flow statements fall into three categories:

1. Conditional branching (or) Decision Making or Non-iterative
2. Looping or iterative or repetitive
3. Unconditional branching.

14.3. Conditional Branching (or) Decision Making:

Conditional branching is the most basic control feature of any programming language. It enables a program to make decisions, to decide whether or not to execute a statement or a block of statements based on the value of an expression. The expression may result in either true or false value. Since the value of the expression may vary from one execution to another, this feature allows a program to react dynamically to different data.

C supports various types of conditional branching statements. The following categories illustrate several conditional control structures.

- **Simple if**
- **if ..else**
- **else if ladder**
- **Nested if**
- **Switch**

14.3.1. Decision Making with Simple if statement:

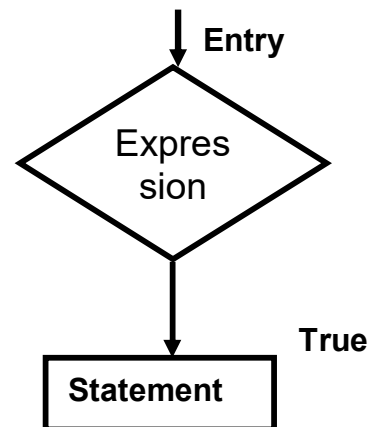
The **simple if** statement is wonderful decision making statement and is used to control the flow of execution of a single or multiple instructions.

The general form of “**simple if**” follows:

```
If (condition/expression)  
Statement;
```

In this statement the given condition is tested first and responds accordingly. If the result of expression is true then the given statement is executed. If the result is false the statement cannot be executed.

When multiple statements are to be executed using if control structure then it may be referred as compound if.



Syntax:

```

    if (expression)
    {
        statement-block;
    }
    statement-x;

```

The statement-block may be a single statement or a group of statements. If the expression is true statement-block will be executed, other wise the statement-block will be skipped and the execution will jump to the statement-x.

Ex:Program to find biggest of two numbers.

```

main()
{
    int a, b;
    printf("\n\t Enter A value          : ");
    scanf("%d", &a);
    printf("\n\t Enter B value          : ");
    scanf("%d", &b);
    if (a>b)
        printf("\n %d is Greater than %d", a, b);
    if (b>a)
        printf("\n %d is Greater than %d", b, a);
}

```

14.3.2.Decision making with if – else statement:

In **if-else** control statement there exists an extension of the simple if statement. It allows the user to perform another block of statements in case the condition result is false.

syntax :

```

if (expression)
    statement-x;
else
    statement-y ;

```

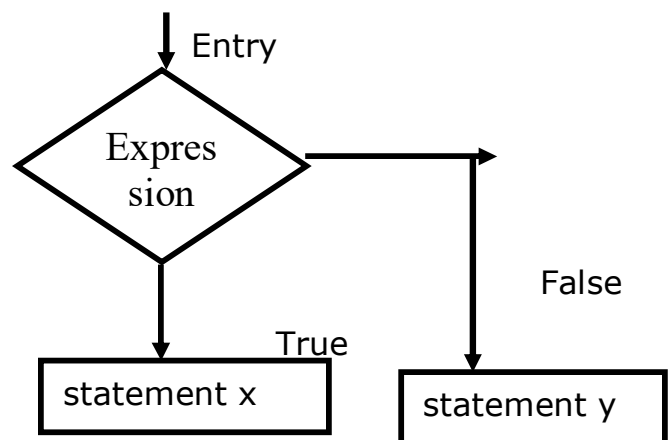
Here the **expression** is evaluated; if the result of the expression is a true then statement-x is executed otherwise statement-y will be executed.

Ex:**Program to check whether given number is even or odd**

```

main()
{
    int n;

```

**Flow graph**


```
printf("\n Enter a number..:");
scanf("%d",&n);
if (n%2==0)
    printf("\n Given number is even");
else
    printf("\n Given number is odd");
getch();
}
```

14.3.3.Decision making with else.. if ladder:

In **else..if ladder** number of conditions are checked depending on the falsity of the previous condition. Literally, too many conditions are evaluated in if..else ladder.

Syntax:

```
    If <condition1>
    {
        -----
    }
    else if <condition2>
    {
        ----- True block 1
    }
    else
    {
        ----- False block
    }
```

In this, condition1 is checked and if it is true then its corresponding condition is executed. If the condition is false then next condition is verified. If all the given conditions are false then false block is executed. Only one of all the available blocks gets executed. After the execution of any one of the blocks, control is transferred to next statement after the construct.

Ex:

program to find biggest of three numbers

```
main()
{
    int a,b,c;
    clrscr();
    printf("enter three numbers:");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
        if(a>c)
            printf("%d is big",a);
        else
            printf("%d is big",c);
}
```

```
        else if(b>c)
            printf("%d is big",b);
        else
            printf("%d is big ",c);
    }
```

Program to award grade to a student

```
main()
{
    int comp,mat,stat,tot;
    float avg;
    char gra;
    printf("\n\t Enter Computers marks  : ");
    scanf("%d", &comp);
    printf("\n\t Enter Maths marks  : ");
    scanf("%d", &mat);
    printf("\n\t Enter Mathematics marks  : ");
    scanf("%d", &stat);
    tot = comp+maths+stat;
    avg = tot / 6.0;
    if(comp>=35 && maths >= 35 && stat>=35)
    {
        if (avg >= 80)
            gra = 'A';
        else if (avg >= 70)
            gra = 'B';
        else if (avg >= 60)
            gra = 'C';
        else if (avg >= 50)
            gra = 'D';
        else
            gra = 'E';
    }
    else
        gra = 'F';
    printf("\n|          JKC COLLEGE, GUNTUR          |");
    printf("\n-----");
    printf("\n\n\tComputers :%d",Comp);
    printf("\n\n\tMaths: %d",maths);
    printf("\n\n\tStatistics: %d",stat);
    printf("\n-----");
    printf("\n\n\tTotal Marks      :%d",tot);
    printf("\n\n\tAverage      : %8.2f",avg);
    printf("\n-----");
    printf("\n\n\t\t\tGrade awarded   : %c",gra);
    printf("\n\n-----");
}
```

14.3.4. Decision making with Nested if:

A **nested if** control structure consists of multiple if statements in one another. Here each if statement consists of subsequent branching statement. Literally a nested if consists of one if statement in another if statement. It is used when multiple conditions are to be evaluated.

Syntax:

```
if(expression)
{
    if(expression)
    {
        if(expression)
        {
            ---
            ---
        }
    }
}
```

Here evaluations of expressions or conditions are based on the first condition. If the first condition itself is false, then there is no way of evaluating other conditions. At any level of expression the program control may be altered.

Ex:

Program Biggest of 3 numbers using nested if

```
main()
{
    int a,b,c,big;
    printf("\n Enter the value of a : ");
    scanf("%d",&a);
    printf("\n Enter the value of b : ");
    scanf("%d",&b);
    printf("\n Enter the value of c : ");
    scanf("%d",&c);
    if (a>b)
        if (a>c)
            big = a;
        else
            big = c;
    else
        if (b>c)
            big = b;
        else
            big = c;
    printf("\nBiggest of three numbers is:%d",big);
}
```

14.3.5. Switch Statement:

C provides a special kind of conditional control structure that acts as an alternative to **if..else ladder**. When there are more conditions or paths in a program, if-else branching can become more difficult. In such situations switch may act better. The **switch** statement allows the user to specify

an unlimited number of execution paths based on the value of a single expression. Each execution path is referred as a case.

However, all the cases should be unique.

Each case must be terminated by a **'break'** statement. The **'default'** case is not mandatory.

In a switch statement, there are four different keywords to be used:

- **switch**
- **case**
- **break**
- **default**

Though the switch control structure enables the user to improve clarity of the program, it causes more errors. So, it requires more attention while implementation.

Syntax:

```
switch(expression)
{
    case value1:
        statement;
        break;
    case value2:
        statement;
        break;
    :
    :
    :
    default :
        statement;
}
```

Among all the cases, only one case can be executed successfully because each case is terminated by a **'break'** statement.

Ex:

Program to accept two integer values and perform arithmetic operation by getting the user input.(1) Addition,2) Subtraction ,3) Multiplication,4) Division. 5) Exit).

```
main()
{
    int a, b, c, ch;
    clrscr();
    printf("\n\t\t\t Enter two numbers      :      ");
    scanf("%d %d", &a, &b);
    printf("Enter your choice:")|
    printf("(1)Addition\n2)Subtraction");
    printf("\n3)Multiplication");
```

```
printf("\n4) Division. \n5) Exit").
scanf("%d",&ch);
switch (ch)
{
case 1:
    c = a + b;
    break;
case 2:
    c = a - b;
    break;
case 3:
    c = a * b;
    break;
case 4:
    c = a / b;
    break;
default :
    printf("\n Invalid option ");
    exit(0);
}
printf("\n\t\t\t Result   :%d",c);
}
```

14.4.Looping structures:

Some times, in a program, a statement or a block of statements need to be executed repeated number of times. In such situations decision control structures may not be useful, as they do not transfer the control back. Hence the user may require another form of control structures, which perform a group of instructions for a fixed number of times. Such control structures are named as looping control structures. C language provides three different iterative or looping structures.

- **while** loop
- **do...while** loop
- **for** loop

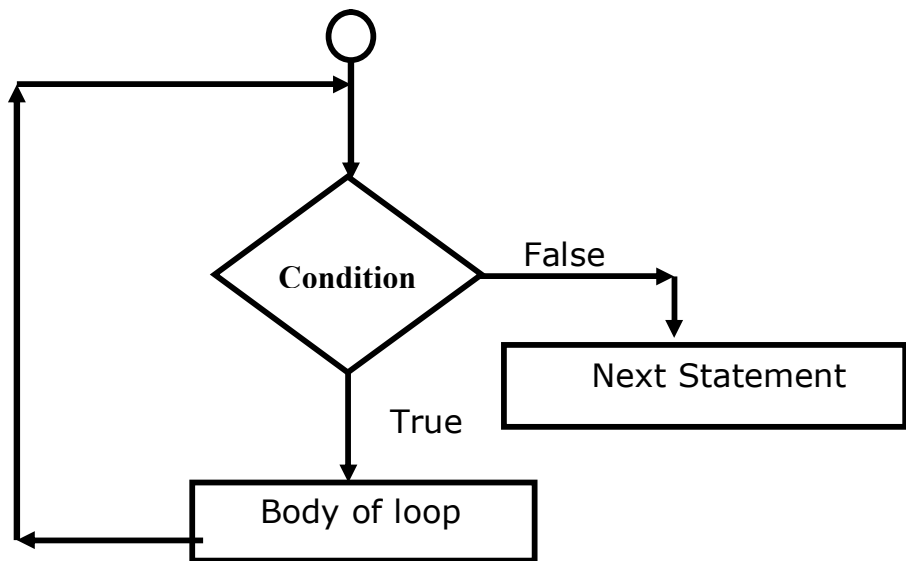
14.4.1.While statement:

The **while** control structure executes a single or multiple statements for repeated number of times based on a given condition. It executes the statements as long as the given condition or expression results in a true value. It terminates execution as and when the condition is false.

Syntax: initialization statement;
 while(condition)
 {
 ...
 Condition reachable statement;

}
Flow graph:

Here the condition is tested every time, it executes the block of statements. The keyword **while** verifies the trueness and falsity of the expression and responds accordingly. If the condition is false for the first time the minimum number of iterations is 0 in while control structure. It requires three statements in order to perform repetitive tasks.



They are

- Initialization statement
- Conditional statement
- Condition reachable statement

If any of the above statements is ignored then the while may not perform well.

Ex:

Program to print the numbers from 1 to 10

```

main()
{
    int i;
    i=1;
    while (i<=10)
    {
        printf("%d\n", i);
        i++;
    }
}
  
```

Program to print even numbers from 2 to 100

```

main()
{
    int a=2;
    while (a<=100)
    {
        printf("\n %d",a);
        a = a + 2;
    }
}
  
```

```

}

```

Program to print odd numbers from 1 to 100

```

main()
{
    int a=1;
    while (a<=100)
    {
        if (a%2 != 0)
            printf("\n %d",a);
        a++;
    }
}

```

14.4.2.Do - While statement:

C provides another form of while control structure i.e., **do-while** control structure. In **do-while** control structure the statements in the block get executes first, later on the condition is evaluated. Hence the user can assume that the minimum number of iterations for do while control structure as 1, even if the expression or condition results in false for the first time.

Syntax:

Initialization statement;

do

{

Condition reachable statement;

} while(condition);

Flow graph:

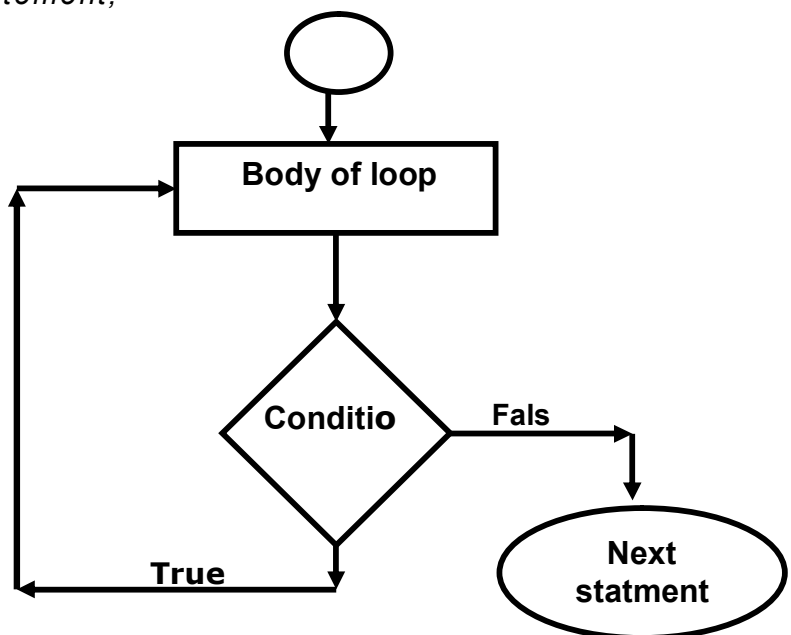
Here the statements in the loop will be executed until the given condition becomes false. The while statement should be terminated by a semicolon (;) in do while.

Ex: print the numbers from 1 to 10

```

main()
{
    int i;
    i=1;
    do
    {
        printf("%d\n", i++);
    }
}

```



```
    } while (i<=10);  
}
```

14.4.3.For Loop:

C provides a more flexible form of looping control structure that improves clarity of the code. It is nothing but for control structure. Usually the for control statement is used to perform fixed number of iterations. The major difference between **for** and **other looping structures** is the number of iterations. In case of while and do-while the number of iterations are indefinite. The user may not predict the number of iterations. On the other hand for specifies the number of iterations in the statement itself.

Syntax:

for(initialization; test condition; increment/decrement part)

```
{  
    Body of the loop;  
}
```

The initialization may contain single or multiple assignment statements. A control variable is involved in this part of statements.

The test condition verifies the validity of the control variable for each iteration.

Increment or decrement part increments or decrements the value of the control variable in order to reach the test condition.

Ex:

program to print the numbers from 1 to 10

```
main()  
{  
    int i;  
    for (i=1 ; i<=10; i++)  
        printf("%d\n", i);  
}
```

14.5.Break and continue Statement:

Break:

This statement takes control out of the switch statement or loop structure. In other words, a break statement takes the control out of the current block in execution. The control is transferred to the statement that follows the block.

Syntax: *break;*

Continue Statement

To skip a part of the body of the loop in execution on certain condition and for the loop to be continued for the next iteration continue statement is used.

Syntax: `continue;`

14.6. goto and label statements:

C supports an unconditional branching statement called 'goto'. This 'goto' is meant for transferring control from one part of the program to another part a label is present. A label is a user-defined word to where the control is supposed to be transferred. The given label must reside in the same function and can appear before only one statement in the same function. Although it may not be preferable to use the goto statement in a highly structured language like C, there may be occasions where the use of goto is desirable.

Syntax:



the following example demonstrates the **goto** statement:

```
void main()
{
    int x = 1;
    abc:
        printf(" %d",x);
        x++;
        if(x <= 5 )
            goto abc;
}
```

14.7.Summary:

This lesson has focused the major elements of programming - control structures in C language. You have learnt about the conditional, looping and unconditional statements. You have also seen conditional statements: if, if-else, nested If and switch. Also focus on three categories of loops, available in C language: while, do – while and for loop. Usage of break, continue, goto and exit in loops have been covered.

14.8.Technical Terms:

Looping: Repeating a block of statements based on condition or a counter.

Compound Statement: A series of statements, enclosed in curly braces. Compound statements may be nested.

Expression: an expression is a statement in which either arithmetic or assignment operations are performed.

Control statement: A statement, which controls the logical flow of a program. Ex: **if, goto, switch.**

while: It is an entry- controlled loop in which the body of the loop is executed only when the condition is true.

do—while: it is a loop construct, which is used to test a condition after the body of the loop is executed once.

for loop: It is an entry-controlled loop, which provides a more concise loop control.

14.9.Model Questions:

1. Describe the control structures of C language and explain each of them with an example.
2. Discuss various control flow statements available in C language?
3. Explain the various loop constructs in available in C language?
4. How can the do-while loop vary from the while-loop?
5. Explain break and continue statements with example?
6. Explain goto and label statements with an example?

14.10.References:

Byron C. Gottfried “**Programming with C**”

Kelly & Pohl “ **A book on C** “

Yeshavanth kanethkar “**Let us C**”

Ravi Chandran “**Programming in C**”

Yeshavanth kanethkar “**Working with C**”

Blaguruswamy “**ANSI C**”

- G.RAMANJIAH.,M.Sc(Computers)

Lecturer, Dept.Of Computer Science,

JKC College, GUNTUR.

Lesson - 15

FUNCTIONS AND STORAGE CLASSES

15.0 Objectives:

The main objectives of this lesson are:

- To understand what a function is
- To know the advantages of functions
- To understand different types of functions
- To understand the function declaration, variable declaration and function calling in a program
- To understand the concept of recursion
- To understand the concept of storage classes

Structure of the Lesson:

- 15.1. Introduction
- 15.2. Functions
- 15.3. Advantages of functions
- 15.4. Types of functions
- 15.5. Types of function calls
- 15.6. Recursion
- 15.7. Storage classes
 - 15.7.1. Auto
 - 15.7.2. Static
 - 15.7.3. Extern
 - 15.7.4. Register
- 15.8. Summary
- 15.9. Technical terms
- 15.10. Model questions
- 15.11. References

15.1. Introduction:

A large program may be split into several sub programs in order to reduce complexity. Bigger applications are not manageable as they contain large code. Modular programming is the concept of splitting a problem or program into several manageable pieces called sub modules or sub programs. C provides such modular programming in an extensive manner.

15.2.Functions:

A **function** is a self-contained block of statements to perform a sub task in a program. It can be referred as a module or procedure or a subtask. A function itself is not a program; rather it extends the ability of a program.

As C is a function dependent language, it supports a rich set of functions in a sophisticated way. Every C program starts with at least one function called **main()**. There are two different categories in functions, system defined and user defined functions.

System defined functions:

A system-defined function is a subprogram, which is prewritten by the compiler. There is a library in C for system defined functions. For example, printf(), scanf(), clrscr(), getchar() etc.,

User Defined Functions:

Some times, the user may require a function, which performs a specific task of his own. In such cases, C permits user-defined functions.

Every function has its own significance and provides a separate scope for variables. A function requires three different types of statements to be specified by the user. They are

- Function prototype declaration
- Function Definition
- Function calling

As C supports top down approach in executing the programs, explicit declaration of the functions must be required in a program.

Every function requires definition without which the system is able to perform nothing. Literally the user has nothing to do without defining any function.

Unless the user makes an explicit call to the function, the function declaration & definition have no significance and they perform nothing.

Hence the above three statements are essential for implementing functions.

15.3. Advantages of Functions:

1. Program debugging is made easy if a C program contains functions.
2. Functions allow a larger task to be subdivided into several smaller tasks, so that they can be managed easily.
3. The length of the source program can be reduced to a maximum extent using functions.
4. The same function can be used for many programs once it is written.
5. Functions may increase program execution speed.
6. Functions improve optimum utilization of memory.
7. Functions are more reliable.

Structure of a Function:

```
return type function - name (data type arg1, data type arg2.....)
{
    local variable declarations ;
    // body of function
    .....
    return(expression);
}
```

return type is nothing but specifying the type of the value being returned by the function to its parent function or caller function.

Every function requires a name (function name) so that it can be referred or identified by the user. Function names are unique in a program. While specifying names, make sure that the name doesn't possess any space in between the characters.

A function may accept any number of values as input from the caller function in order to perform the task assigned to it. Each value is referred as an **argument** or **parameter**. Usually parameters are specified in the brackets '(', ')'.

The keyword return is not mandatory. It is the last statement of a function. If at all, a function wants to send a value back to the caller, return statement may be kept in use.

15.4.Types of Functions:

In C, functions are divided into the following categories:

- System Defined Functions
- User Defined Functions

Both system defined and user defined functions may fall in the following category:

- Functions with no arguments and no return type.
- Functions with arguments and no return type.
- Functions with arguments and return type.
- Functions with no arguments and return type.

Functions with no arguments and no return type:

In this category of function, the caller function does not send any argument to the function and the called function does not return any value.

Functions with arguments and no return type:

In this category of function, the caller function sends one or more values to called function but in return the called function does not return any value.

Functions with arguments and return type:

In this category of function, the caller and called functions send values to one another.

Functions with no arguments and return type:

This category of function doesn't accept any value from the caller function but returns one value back to the caller.

15.5.Types of Function calls:

There are two types of function calls.

1. Call by value
2. Call by reference.

Call by value:

In call by value, the actual parameters are sent to the called functions from the caller.

Call by reference:

In call by reference, instead of copy, the address of the actual parameter is sent to the called function from caller. Therefore, the called function can alter the value of the actual parameters. In this case, pointers are used.

Ex: **Write a function to clear the screen.**

```
main()
{
    void screenclear();
    screenclear();
    getch();
}
void screenclear()
{
    int i,j;
    for(i=1;i<=24;i++)
    {
        for(j=1;j<=80;j++)
        {
            gotoxy(j,i);
            printf(" ");
        }
    }
    gotoxy(1,1);
}
```

Write a program to find factorial of a given number.

```
main()
```

```
{
    void fact();
    clrscr();
    fact();
    getch();
}
void fact()
{
    int x;
    long int f=1;
    printf("\n Enter a number to find factorial : ");
    scanf("%d",&x);
    while(x>0)
    {
        f = f * x;
        x--;
    }
    printf("\n Factorial of given number is : %ld",f);
}
```

15.6.Reursion:

When a called function in turn calls another function, then a process of chaining occurs. Recursion is special case of this process. In other words, **Recursion** is a process by which a function calls itself repeatedly, until some specified condition is true. The number of recursive calls is limited to the size of the stack. A function is called 'recursive' if a statement within the body of a function calls the same function. Sometimes called 'circular definition'.

Ex:

```
void ramu()
{
    static count = 1;
    printf ("%d", count);
    count++;
    ramu();
}
main ()
{
    ramu ();
}
```

What will the program do? First it prints the value of count, which is 1 then it increments count; then it calls itself. The second time, count equals 2. This repeats the function infinite times. The output will be

1 2 3 4 5.

At some point, the computer will run out of stack memory, and the program will abort with a runtime error. This illustrates an important point about recursive programming. You must include a stop point or the program will run forever (or until it runs out of memory). So, we can modify the above function so that calls itself only three times.

```
void ramu ( )
```

```
{
    static count = 1;
    if ( count > 3)
        return;
    else{
        printf("%d\n", count);
        count++;
        ramu ( );
    }
}
main ( )
{
    ramu ( );
}
```

Each time the function is called, new storage is allocated for the parameters for the **auto** and **register** variables so that their values in previous, unfinished calls are not overwritten. Parameters are directly accessible to the instance of the function in which they are created. Previous parameters are not directly accessible to ensuing instances of the function.

Note that the program would not end if count were automatic rather than fixed because it would dynamically create a new variable called count and reinitialize it to 1 with each call. For each new call, the compiler creates a whole new set of automatic variables. Even though they have the same name, they refer to different memory areas. That variables declared with **static** storage do not require new storage with each recursive call. Their storage exists for the lifetime of the program. Each reference to such a variable accesses the same storage area.

15.7.Storage classes:

Every variable and functions in C is associated with a data type.

We have already seen that all variables have an attribute, its type. In addition to the data type, C associated a storage class with variables and functions. Storage class is pertaining to the storage of the variable.

A storage class is associated with its object certain properties such as scope and lifetime in a program. By scope of an object we understand that the portions of the program in which the object can be used. The lifetime of an object is the period of time during which it is allocated storage space in memory.

The scope and lifetime of objects are determined by storage class specifications. There are four types of storage classes.

- Auto
- Register storage class
- Static
- Extern

15.7.1.Auto variables:

Auto is the default storage class to be used within a program or file. For ex:

```
int x; (or) auto int x;
```

Here, The 'auto' keyword declares the storage specifier for the variable x is automatic.

The scope of automatic variable is limited to the block in which it appears. They are created when the function is called and destroyed when the function is exited. Their memory locations get deallocated when the function is existed. They are recreated when function is called again.

Automatic variables are not initialized to any value when they are created. They usually start with garbage values. Automatic variables have two pleasing aspects. First, memory space used economically. Second their local scope protects us from affecting them inadvertently in other function. So, variable in other function need not necessarily be given different names.

Ex: Auto variables

```
void main()
{
    auto a=1;
    {
        auto a=2;
        {
            auto a=3;
            printf("%d \t",a);
        }
        printf("%d\t",a);
    }
    printf("%d\t",a);
}
```

Output: 3 2 1

15.7.2.Static variables:

The static variables are defined within a function. They have the same scope of the rules of the automatic variables, but in the case of static variables the contents of the variables will be retained thought the program. Static is a keyword used to define the storage class as static. In the absence of any initialization, static variables are initialized to zero.

Like automatic variables, static variables are also local to the function. When they are declared.

Ex:

```
int myblock()
{
    static float a;
    static int b[]={1,2,3,4,5};
    -----
}
```

The variable a and the array b have been declared static. They will start with an internal value and their value is retained when the function is exited.

15.7.3. External variables:

The variables that are declared before the function main() are globally existing for all the functions inside the program. These variables are called global variables. The extern declaration of the variables makes the variables to be available even for the external functions that are called from the program.

The extern storage class does not create a variable. It only informs the compiler of its existence and so the extern declaration cannot include any initialization.

```
Extern int I;
```

Here, I is a global variable which may be defined in another file, thus supporting multi-file communication.

External variables are declared outside the block in which it is used. It can be declared either before the block or after the block. If it is declared after the block, we have to declare the variable as extern before using it. If the variable is declared before the block it is not necessary to declare it as extern.

```
Ex:# include<stdio.h>
    int a,b;
    float x;
    void main()
    {
        -----
        -----
    }
```

Here a, b and x are defined outside of all functions, including main. Therefore they are external. However, the word extern should not precede them. They are recognized as external by virtue of their position in the program.

15.7.4. Register variables:

The keyword register is used to specify the storage class of the variables. All the computers have a few memory locations on the microprocessor. These memory locations are known as registers. Storage and retrieval of data from a register can be done by a program much faster than from a location in the main memory of the computer.

The register specification is quite useful when a particular value is frequently required, for example, index variables. Otherwise it is stored in the RAM with the default storage class.

```
Ex: Register int a;
```

The scope and life of a register variable is similar to that of the automatic variables. There are some restrictions on register variables. Since a machine register is usually single word, many compilers only allow those variables that fit into the word to be placed in registers.

15.8. Summary:

In this lesson you have learnt what is a function, how to use a functions, how the functions interact with one another, how to send them information using arguments, and how to use them to return values.

In this lesson we have discussed the concepts of storage class and scope for variables in a C program. The language provides four storage classes: automatic, register, external, and static.

15.9. Technical Terms:

Function: In computer science, a subroutine (function, procedure, or subprogram) is a sequence of code, which performs a specific task.

Local variable: A variable that exists only inside a particular function, not affecting anything outside the function.

Global variable: A variable that can be accessed by all parts of a program.

Register: Memory device that is the part of computer memory that has a specific address and that is used to hold information of a specific kind.

Secondary storage: Secondary storage is a category of computer storage. It is used to store data that is not in active use. It is usually slower and has higher capacity than primary storage, and is almost always non-volatile.

Recursion:When a function calls itself, either directly or indirectly. If this isn't clear, refer to the entry for "recursion."

15.10. Model Questions:

1. What is a function? Explain advantages of functions in a program?
2. Write short notes on different types of functions?
3. Explain recursion with an example?
4. Write about storage classes?

15.11. References:

Byron C. Gottfried “**Programming with C**”

Kelly & Pohl “**A book on C**”

Yeshavanth kanethkar “**Let us C**”

Ravi Chandran “**Programming in C**”

Yeshavanth kanethkar “**Working with C**”

Blaguruswamy “**ANSI C**”

- **G.RAMANJIAH.,M.Sc(Computers)**
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson- 16**ARRAYS AND STRINGS****16.0 Objectives:**

The main objectives of this lesson are:

- To understand what an array is.
- To understand different types of arrays.
- To know the declaration and initialization process.
- To understand the concept of strings.
- To understand the declaration and initialization of strings.
- To familiarize with string manipulating functions.

Structure of the Lesson:

- 16.1. Arrays**
- 16.2. Single dimensional arrays**
- 16.3. Two dimensional arrays**
- 16.4. Multi dimensional arrays**
- 16.5. Strings**
- 16.6. String handling functions**
- 16.7. Summary**
- 16.8. Technical terms**
- 16.9. Model questions**
- 16.10. References**

16.1.Arrays:

An array is a group of related data items that shares a common name (or) an array is a collection of identically (similarly) typed variables stored contiguously in memory.

Each variable in an array is called an element and can be accessed by giving the array name plus an index expression called a subscript. A subscript value of 0 identifies the initial element; a value of 1 identifies the next element, and so forth.

The most basic purpose of arrays is to store large amounts of related data that share the same data type. Suppose you want to store 100 names of your relatives and friends. This requires 100 memory locations. Obviously, it would be extremely tiresome to declare 100 variables, each with a unique name. Arrays provide a solution to this problem.

16.2.Single dimensional arrays:

Apart from the broad spectrum of data types C supports arrays. This data type is useful when a group of elements are to be represented by a common name.

An array is a group of elements that share a common name, that are differentiated from one another by their positions within the array. An array is a collection of homogeneous data.

Declaring arrays: -

An array must be declared, since it is basically a variable.

Syntax: *type variable-name [size];*

Every element in the array is manipulated using its index. The starting index of element is 0 and ends with n-1. A list of data items can be given one variable name using one subscript and such a variable is called one-dimensional array.

➤ Declaring an Array

You can declare an array by placing a pair of brackets after the array name. To specify the size of an array, enter the number of elements within the brackets.

Syntax

Type variable_name[size]

int a[10];

Represents ten integers, this complete set is called an *array*.

Here, *int* specifies the type of the variable, just as it does with ordinary variables and the word **a** specifies the name of the variable. The number **10** tells how many elements of the type *int* will be in our array. This number is often called the “*dimension*” of the array. The bracket [] tells the compiler that we are dealing with an array.

➤ Initializing

We can initialize the elements in array in the same way as the ordinary variables when they are declared. The general form is

Type array_name [size] = {list of values}

For example:

int a[3] = { 1, 2, 3 };

In this form the size may be omitted. In such cases, the compiler allocates enough space for all initialized elements.

For example:

```
Char name[] = { 'S', 'S', 'I', 'T'};
```

Thus the above statement declares the name array of four characters, initialized with the string, **"SSIT"**.

➤ Accessing Elements

You can access the elements inside the array by using array name and index of the element to be accessed. All the array elements are numbered, starting with 0. Thus **a[2]** is not the second element of the array but the third. In our program we are using the variable *i* as a subscript to refer to various elements of the array. This variable can take different values and hence can refer to the different elements in the array in turn. This ability to use variable as subscripts is what makes arrays so useful.

You can access the first element by using **a[0]**, second element **a[1]**, and so on.

```
I.e., a[0] = 11  
a[1] = 21  
a[2] = 31
```

Ex: Program to accept 5 elements into an array and display them back.

```
main()  
{  
    int a[5],i;  
    for(i=0;i<5;i++)  
    {  
        printf("\n Enter the value of a[%d] : ",i);  
        scanf("%d",&a[i]);  
    }  
    printf("\n");  
    for(i=0;i<5;i++)  
    {  
        printf("\t%d",a[i]);  
    }  
}
```

Program to find sum of elements of an array with n elements.

```
main()  
{  
    int a[50], i, n, sum = 0;  
    do  
    {  
        clrscr();
```

```
        printf("\n Enter number of elements    :  ");
        scanf("%d",&n);
    } while (n>50 || n<=0);
    for(i=0;i<n;i++)
    {
        printf("\n Enter the value of a[%d]    :  ",i+1);
        scanf("%d",&a[i]);
        sum += a[i];
    }
    clrscr();
    for(i=0;i<n;i++)
    {
        printf("\t%d",a[i]);
    }
    printf("\n\n Sum of elements    :    %d",sum);
    getch();
}
```

Program to find biggest of elements of an array.

```
main()
{
    int a[50],i,n,big;
    do
    {
        clrscr();
        printf("\n Enter number of elements    :  ");
        scanf("%d",&n);
    } while (n>50 || n<=0);
    for(i=0;i<n;i++)
    {
        printf("\n Enter the value of a[%d]    :  ",i);
        scanf("%d",&a[i]);
    }
    big = a[0];
    for(i=1;i<n;i++)
    {
        if (big<a[i])big=a[i];
    }
    for(i=0;i<n;i++)
    {
        printf("\t%d",a[i]);
    }
    printf("\n\n Biggest of elements    :    %d",big);
    getch();
}
```

16.3. Two-dimensional arrays:

So far we have discussed the array variables that can store a list of values. There will be situations where a table of values will have to be stored.

C allows us to define such tables of items by using two-dimensional arrays. This type of arrays can be declared as

Type array_name[row-size][column-size];

For example

int a[3][3];

This two dimensional array consists of three rows and e three columns.

➤ ACCESSING THE ELEMENTS

The first row first element can be accessed by `a[0][0]`, the first row second element can be accessed by `a[0][1]`, similarly the third row second by `a[2][1]` and soon.

Multidimensional arrays are stored in row-major order, which means that the last subscript varies fastest.

➤ INITIALIZATION

Like single dimensional arrays the initialization can be done in multidimensional arrays. When initializing a multidimensional array, you may enclose each row in braces. If there are too few initializes, the extra elements in the row are initialized to zero. Consider the following example:

```
int m_arr [5] [3] = {{1,2,3},  
                    {4},  
                    {5, 6, 7}};
```

This declares an array with five rows and three columns, but only the first three rows are initialized, and only the first element of the second row is initialized.

```
1 2 3  
4 0 0  
5 6 7  
0 0 0  
0 0 0
```

if we do not include the inner brackets, as in

```
int m_arr[5] [3] = {1, 2, 3,  
                  4,  
                  5, 6, 7};
```

the result is


```

1 2 3
4 5 6
7 0 0
0 0 0
0 0 0

```

Ex Program to accept an m * n matrix and display it.

```

main()
{
    int a[10][10],i,j,m,n;
    printf("\n\n\n\t\tEnter total number of rows    : ");
    scanf("%d",&m);
    printf("\n\n\n\t\tEnter total number of colomns  : ");
    scanf("%d",&n);
    for (i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("\n Enter the element a[%d][%d] : ",i+1,j+1);
            scanf("%d",&a[i][j]);
        }
    }
    printf("\n\n Given Matrix is....\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("\t %d",a[i][j]);
        }
        printf("\n");
    }
}

```

16.4. Multi- dimensional arrays:

C allows arrays of three or more dimensions. The exact limit is determined by the compiler. A subscript expression can also have multiple subscripts, as follows:

Type array_name[expression1][expression2]....[expression n];

Subscript expressions associate from left to right. The leftmost subscript expression, *expression1*, [*expression2*], is evaluated first. The address that results from adding *expression1* and *expression2* forms a pointer expression; then *expression3* is added to this pointer expression to form a new pointer expression, and so on until the last subscript expression has been added. The indirection operator (*) is applied after the last subscripted expression is evaluated, unless the final pointer value addresses an array type (see examples below).

Expressions with multiple subscripts refer to elements of “multidimensional arrays”. A multidimensional array is an array whose elements are arrays. For example, the first element of a three-dimensional array is an array with two dimensions.

16.5. Strings:

A string is a group of characters, usually letters of the alphabet. In order to format your printout which has meaningful names and titles, you need the ability to output text data. Actually you have already been using strings, the second program in this tutorial, way back in Chapter 2, output a message that was handled internally as a string. A complete definition of a string is a series of **char** type data terminated by a null character.

When C is going to use a string of data in some way, either to compare it with another string, output it, copy it to another string, or whatever, the functions are set up to do what they are called to do until a null, which is a zero, is detected. Such a string is often called an ASCII-Z string. We will use a few ASCII-Z strings in this chapter.

One of the most common uses of arrays is to store strings of characters. A string is an array of characters terminated by a **null character**.

A null character is a character with a numeric value of zero. It is represented in c by the escape sequence ‘\0’. A **string constant**, some times called a **string literal**, is any series of characters enclosed in double quotes. It has a data type of **array of char**, and each character in the string takes up one byte. In addition, the compiler automatically appends a **null character** to designate the end of the string.

Declaring and initializing strings

To store a string in memory, you need to declare an array of type char, you may initialize an array of chars, with a string constant. For example

```
char str[] = “some text”;
```

The array is one element longer than the number of characters in the string to accommodate the trailing null character. str[], therefore, is ten characters in length. If you specify an array size, you must allocate enough characters to hold the string. In the following example, for instance, the first four elements are initialized with the characters ‘y’, ‘e’, ‘s’, and null ‘\0’. The remaining six elements receive the default initial value of zero.

```
char str[10] = “yes”;
```

For example:

```
char code[ ] = “abc”;
```

Initializes code as a four-element array of characters. The fourth element is the null character, which terminates all string literals.

The declaration

```
char s[] = "abc", t[4] = "abc"; is identical to  
char s[] = {'a', 'b', 'c', '\0'}, t[4] = {'a', 'b', 'c'};
```

If the string is shorter than the specified array size, the remaining elements of the array are initialized to 0.

➤ Reading and Writing Strings

You can read and write strings with the `printf ()` and `scanf ()` functions by using the `%s` format specifier. For `scanf ()`, the data argument should be a pointer to an array of characters that is long enough to store the input string. The input string is terminated by any space character. After reading in the input characters, `scanf ()` automatically appends a null character to make it a proper string. On the `printf ()` side, the data argument should be a pointer to a null-terminated array of characters. `Printf ()` outputs successive characters until it reaches a null character.

Ex: Program to accept a string and count no.of characters.

```
main()  
{  
    char s[20];  
    int i,charcount=0;  
    clrscr();  
    printf("\n Enter any string : ");  
    gets(s);  
    for(i=0;s[i]!='\0';i++)  
    {  
        charcount++;  
    }  
    printf("\n No. of characters in the given string : %d",charcount);  
    getch();  
}
```

A program to check whether a given string is palindrome or not.

```
main()  
{  
    char a[20],b[20];  
    int i,l=0,flag=0;  
    printf("\n enter any string : ");  
    gets(a);  
    l = strlen(a);  
    for(i=0;i<l;i++)  
    {  
        b[i]=a[l-(i+1)];  
    }  
    for(i=0;i<l;i++)
```

```

    {
        if(b[i]!=a[i])
            flag=1;
    }
    if(flag==1)
        printf("\n %s is not a palindrome.",a);
    else
        printf("\n %s is a palindrome.",a);
}

```

Write a program to count number of vowels in a given string.

```

main()
{
    char s[20];
    int i,l=0;
    printf("\n enter any string:");
    gets(s);
    for(i=0;s[i]!='\0';i++)
    {
        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u' || s[i]=='A' || s[i]=='E' || s[i]=='I' ||
           s[i]=='O' || s[i]=='U' )
            l++;
    }
    printf("\n No. of vowels in a given string are : %d",l);
    getch();
}

```

16.6. String handling functions:

A string in C is simply an array of characters. The following are the basic string handling functions. All the string-handling functions are prototyped in: `#include <string.h>`

strcpy	Copy one string into another.
strcmp	Compare string1 and string2 to determine Alphabetic order.
strcpy	Copy string2 to string1.
strerror	Get error message corresponding to specified error number.
strlen	Determine the length of a string.

strncat	Append n characters from string2 to string1.
strncmp	Compare first n characters of two strings.
strncpy	Copy first n characters of string2 to string1 .

strlen() Function :

This function counts the number of characters present in a string.

Ex:

```
main()
{
    char s[20];
    int i,charcount=0;
    clrscr();
    printf("\n Enter any string : ");
    gets(s);
    charcount = strlen(s);
    printf("\n No. of characters in the given string : %d",charcount);
    getch();
}
```

strcpy() Function :

This function copies the contents of one sting into another. The base address of the source and target strings should be supplied to this function.

Ex:

```
main()
{
    char a[20],b[20];
    int i;
    clrscr();
    printf("\n Enter any string : ");
    gets(a);
    strcpy(b,a);
    printf("\n Copied string is : ");
    puts(b);
    getch();
}
```

strrev() Function :

This function reverses the string.

Ex:

```
main()
{
    char a[20],b[20];
    int i,stringlength=0;
    clrscr();
    printf("\n Enter any string : ");
    gets(a);
    strrev(a);
    printf("\n Reversed string is : ");
    puts(a);
}
```

strcat() Function :

This function concatenates the source string into target string. It is necessary to place '\0' into the target string, to make its end.

Ex:

```
main()
{
    char a[20],b[20],c[40];
    int i,j;
    clrscr();
    printf("\n Enter 1st string:");
    gets(a);
    printf("\n Enter 2nd string :");
    gets(b);
    strcpy(c,a);
    strcat(c,b);
    printf("\n concatenated string is  ");
    puts(c);
}
```

16.7. Summary:

In this Chapter, we have introduced one form of compound data type: the **array**. An array is a block of a number of data items all of the same type allocated in contiguous memory cells.

We have seen that, in C, an array may be declared using the syntax: -specifier
identifier>[<size>];
specifying the type of the data items, and the number of elements to be allocated in the array.

This lesson has discussed a very common data type in C programs: the string. We have briefly introduced the concept of an *abstract data type* as consisting of a data declaration and a set of

operations on data items of that type. We have defined a user-defined type, STRING, for string data and used it throughout the chapter.

We have described the functions for string manipulation, strlen() and strcpy() as well as string operation, strcmp() and strcat(), have been described. Other functions described include atoi(), strncmp(), and strncpy(). Throughout the chapter we have shown numerous examples of programs for string processing.

16.8. Technical Terms:

Array: Arrays are a data structure that is used to store a group of objects of the same type sequentially in memory. All the elements of an array must be the same data type.

Character Array. A set of elements of type char. (Can be used to store a string).

Pointer: pointer is a variable, which contains the address of another variable.

String: a string is a sequence of characters terminated by a null character.

Concatenation: This term is used to describe the combining of strings

16.9. Model Questions:

1. What is an array? Explain different types of arrays?
2. Write a program to read and display a two dimensional array?
3. Explain string-handling functions in C?
4. Write a program to check whether a given string is palindrome or not.

16.10. References:

Byron C. Gottfried “**Programming with C**”

Kelly & Pohl “**A book on C**”

Yeshavanth kanethkar “**Let us C**”

Ravi Chandran “**Programming in C**”

Yeshavanth kanethkar “**Working with C**”

Blaguruswamy “**ANSI C**”

- **G.RAMANJIAH.,M.Sc(Computers)**
Lecturer, Dept.Of Computer Science,
JKC College, GUNTUR.

Lesson- 17

POINTERS

17.0 Objectives:

The main objectives of this lesson are:

- To understand what a pointer is.
- To know the abilities of pointers in C language.
- To understand pointer variable declaration and its usage in a program.
- To know the pointer usage in functions and arrays.

Structure of the Lesson:

17.1. Introduction

17.2. Pointer

17.3. Abilities of pointers

17.3. Summary

17.4. Pointers in functions

17.5. Function pointers

17.6. Different kinds of pointers

17.7. Summary

17.8. Technical terms

17.9. Model questions

17.10. References.

17.1.Introduction:

'Pointers' is a wonderful concept provided by C in order to access data in a better way. They provide a convenient environment where in data may be managed by its address locations. Every data that the user stores in memory has certain address location. Address is nothing but a unique ID number associated with every data items placed in memory. In some other way, a pointer provides a convenient way of accessing a data item without referring its name or identifier directly.

17.2.Pointer:

17.2.Pointer:

One of the most powerful features of C, and one that makes it quite close to assembly language, is its ability to refer to the address of program variables. A "pointer declaration" names a pointer variable and specifies the type of the object to which the variable points. *A variable declared as a pointer holds a memory address.*

As a definition, a pointer is a variable that holds the address of another variable. Suppose there is an integer variable, to hold its address it requires an integer pointer. For any floating point value, it requires a float type pointer. So is the case even with other data types.

Pointers enable the users to effectively represent complex data structures. A pointer provides an indirect means of accessing the value of a particular data item. In order to create a pointer, use two different operators like indirection operator (*), address of operator (&).

Syntax:

```
int z=5;
int *p;
p = & z; // address of a.
printf("%d",*p);
```

Here 'p' means address of 'z' and '*p' means value at address p.

Literally, p → address of z;
 *p → value at address(p)

17.3. Abilities of Pointers:

1. To have indirect access to the data.
2. To improve capability of applications.
3. To increase the speed of program execution.
4. To develop hardware level applications.
5. To perform dynamic memory allocation & de-allocation.
6. To return more than one value from a subprogram.
7. A pointer to a pointer may be generated.

Ex:

```
int a, *p1;
float b, *p2;
char c, *p3;
a= 100; b = 23.45; c = 'z';
p1 = & a;
p2 = &b;
p3 = &c;
```

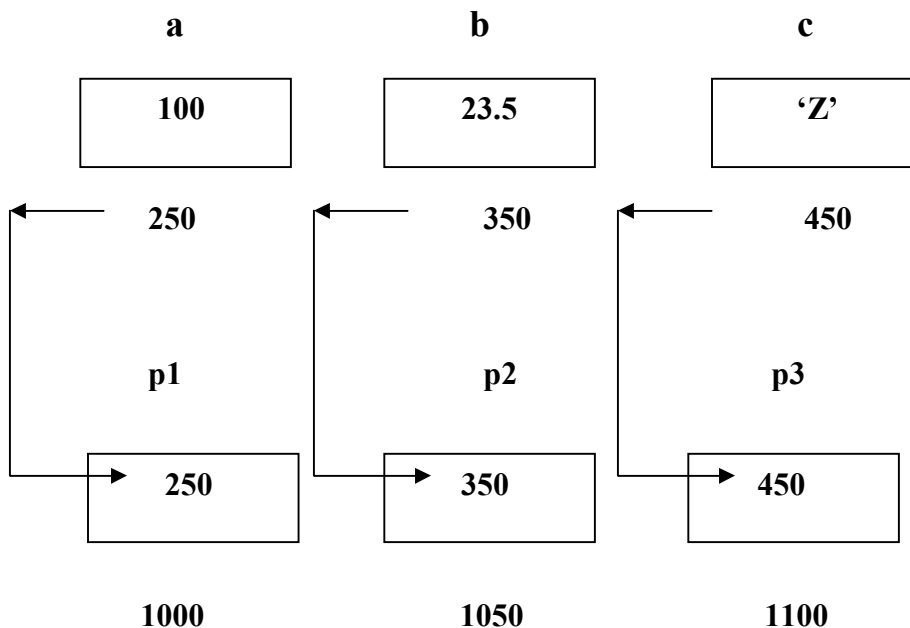
```
printf("%d %u %u %d %d\n", a, p1, &a, *(&a), *p1);
printf("%d %u %u %d %d\n", b, p2, &b, *(&b), *p2);
printf("%d %u %u %d %d\n", c, p3, &c, *(&c), *p3);
```

Output of above program:

```
100 250 250 100 100
23.5 350 350 23.5 23.5
z 450 450 z z
```

The above program prints the address of a, b and c. Here '%u' is the format specifier, '&' it is meant for displaying unsigned integer values. Every address location is an unsigned integer value. Hence you may use %u or %p for pointers.

Lets see how pointers actually work: When you run the above program variables a, b and c are kept in memory as follows:



Here, a, b, c are the different variables and p1, p2, p3 are the pointers. So the pointers hold always the addresses. Moreover, a pointer may have its own address again i.e., a pointer also gets stored somewhere in memory.

→ 100, 23.5, 'Z' are the data items.

→ 250, 350, and 450 are assumed to be the address of a, b, c in memory respectively.

→ 1000, 1050 and 1100 are the addresses of pointers p1, p2 & p3.

In general pointers may perform arithmetic operations like addition and subtraction. Even relational operators may be applied on pointers. However, multiplication and division cannot be performed. Short-hand operators like ++, -- can also be used.

```
p++; //valid
p--; // valid
p-2; // valid
p+3; //valid
a = *p+2; // valid
p * 2; // invalid
p1/p2 ; // invalid
p/3; // invalid
```

Here pointer instrumentations have special meanings. Suppose p is a pointer and it holds address of a variable k; then p+1 means the value of p is incremented by the length of the data type that it points to, such length is called scale factor.

In case of the following data types; length is:

Data type	Length
Int	2
Char	1
Float	4
Long int	4
Double	8

For example, p has the address 2300 for a variable of double type. P+2 means $2300+(2 * 8) = 2317$; p become 2317.

Pointer to Arrays:

A pointer may be applied to arrays. Generally the compiler itself allocates a base address to an array. Moreover, the compiler, in case of arrays, performs implicit addressing. Suppose, we declare an array z as follows:

```
static int z[] = {11,22,33,44,55};
```

Assume that the above is stored in memory as follows:

0	1	2	4	5	index values
11	22	33	44	55	values
					array z:
					p = &z[0]; or p = z; i.e., p = 100.
100	102	104	106	108	addresses

Consider **p** is a pointer to

This implies the address or base address of array **z** into **p**.

Hence you can assume the following representation:

11	22	33	44	55
p+0	p+1	p+2	p+3	p+4

The following program illustrates the pointers to arrays:

```
#include <stdio.h>
void main()
{
    static int a[] = {11,22,33,44,55};
    int *p,i;

    p = a; // assigning address of base address of a

    for(i = 0;i<5;i++)
    {
        printf("%d\t",++(*p));
        p++;
    }
}
```

output:

12 23 34 45 56

Here, ++(*p) means the value at address p is incremented by 1. Hence values 11, 22, 33, 44 and will become 12, 23, 34, 45, 56 respectively.

In some other way, the array elements are represented as follows:

a[0]	a[1]	a[2]	a[3]	a[4]
↓	↓	↓	↓	↓
*(a+0)	*(a+1)	*(a+2)	*(a+3)	*(a+4)

In case of two dimensional arrays, suppose, `int a[2][2] = {11,22,33,44};`

```
a[0][0] → *(*a+0)+0
a[0][1] → *(*a+0)+1
a[1][0] → *(*a+1)+0
a[1][1] → *(*a+1)+1
```

```
Ex: #include<stdio.h>
void main()
{
    int i,j;
    int z[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
            printf ( "%4d",*(a+i)+j) );
        printf("\n");
    }
}
```

```
Output:      1      2      3
              4      5      6
              7      8      9
```

In case of strings or character arrays:

Like one-dimensional array, strings can be manipulated using pointers. A string is always terminated by a NULL character, it becomes easy to manage.

```
Ex: #include<stdio.h>
void main()
{
    char *str = "JKC college";
    char *p;

    while(*p!=NULL)
    {
        printf("%c",*p);
        p++;
    }
}
```

```
Output:      JKC college.
```

17.4. Pointers in Functions:

As 'pointers' is a general concept, it may be used in functions. In case of passing arguments, pointers may be used. In fact, functions: caller and called functions use, pointer in quite a different way.

Functions that call by value may not be useful in certain cases where the reflections of called functions should be done on caller. The following example illustrates the usage of pointers.

```
#include<stdio.h>
void swap(int, int); // prototype declaration

void main()
{
    int a, b;
    a = 5;
    b = 9;

    printf("\n Before swapping....\n");
    printf("%5d%5d",a,b);

    swap(a,b); // called function & is a call by value

    printf("\n After swapping....\n");
    printf("%5d%5d",a,b);
}

void swap(int x, int y)
{
    int temp;

    temp = x;
    x = y;
    y = temp;
    return;
}
```

o/p:

```
Before swapping....
5      9
After swapping....
5      9
```

Here, a, b in main() are the actual parameters and x, y in swap() are the formal or dummy parameters. Though x, y hold values of a, b, the changes made to x, y may not reflect a, b because only the copy of a, b is placed in x, y. Updatons are restricted to the called function only.

If at all, every trivial change is to be reflected in the caller function, then user has to consider 'pointers' concept. The following example illustrates the same.

```
#include<stdio.h>
void swap(int &, int &);

void main()
{

    int a,b;
```

```
a = 5;
b = 9;

printf("\n Before swapping...\n");
printf("%5d%5d",a,b);

swap (&a, &b); // called function & is a call by value

printf("\n After swapping...\n");
printf("%5d%5d",a,b);
}
```

```
void swap(int *x , int *y)
{
    int temp;

    temp = *x;
    *x = *y;
    *y = temp;
    return;
}
```

o/p:

```
    Before swapping....
    5         9
    After swapping....
    9         5
```

17.5. Function Pointer :

Like integer pointer, float pointers and other pointer types, may have function pointers. A function pointer is a pointer that holds the address of a function in memory. It improves program execution speed.

Syntax:

```
Data type (function_ptr>();
```

Example:

```
double (*p>();
```

The following example illustrates function pointer:

```
#include<stdio.h>
```

```
void test(); // proto type declaration.
void main()
{
    void (*p);
```



```
p = test; //assigning address of function to p.

(*p)(); // calling test() using function pointer.
}

void test()
{
    printf(" c ");
    printf("c++");
    printf("vc++");
}
```

17.6. Different kinds of Pointers:

There is a special kind of category in pointers.

- Near Pointers
- Far Pointers
- Huge Pointers

Based on the memory organization, data may be accessed in different ways. Different microprocessors follow different data models. Based on the data models, the above category of pointers will do the work.

Near Pointer:

It takes 2 bytes so it holds the addresses (offset) from 0 to 65535.

Far Pointer:

It takes 4 bytes of memory, it will hold both the segment address (higher order two bytes) and offset (lower order two bytes).

In order to access CGA video memory, we normally use, far pointers.

Huge Pointer:

In this we have another type is called "**Huge Pointer**" which is just like far pointer except that " the address stored in huge pointer will be normalized".

TSR programming or low level programming is performed using Near, far and huge pointers.

17.7. Summary:

In this lesson we have introduced a new data type, a **pointer**. We have seen how we can declare variables of this type using the * and indicating the type of object this variable can point to, for example:

```
int * iptr;
float * fptr;
char * cptr;
```

Declare three pointer variables, `iptr` which can point to an integer cell, `fptr` which can point to a cell holding a floating point variable, and `cptr` which can point to a character cell.

We have also shown how pointer variables may be used to *indirectly* access the value in a cell using the **dereference** operator, `*: y = *iptr;`

As we have seen,, pointers are very useful in developing complex programs. The concept of pointers may be confusing at first, however, a useful tool for understanding the behavior of a program using pointers is to draw the memory picture showing which to cells each pointer is pointing.

17.8. Technical Terms:

Call by reference - passing a pointer to an argument to a function. The function can then change the argument value. See call by value.

Call by value - passing a copy of an argument to a function. The function cannot then change the argument value. C and C++ use call by value argument passing. But also see pointer and reference, also call by reference.

Pointer to data member - a pointer that points at a data member of a class.

Pointer to function - an address of a function or a member function.

17.9. Model Questions:

1. What is a pointer? How it is declared and initialized?
2. What is function pointer? Explain with an example?
3. What are advantages of functions?

17.10. References:

Byron C. Gottfried, “**Programming with C**”

Kelly & Pohl, “ **A book on C** “

Yeshavanth kanethkar, “**Let us C**”

Ravi Chandran, “**Programming in C**”

Yeshavanth kanethkar, “**Working with C**”

Blaguruswamy, “**ANSI C**”

- **G.RAMABJIAH.,M.Sc(Computers)**

Lecturer, Dept.Of Computer Science,

JKC College, GUNTUR.

Lesson :18

BITWISE OPERATORS, STRUCTURES AND UNIONS

18.0 Objectives:

To learn about representing discrete values as bits and elements in a finite set. To appreciate the low-level and system-dependent operations on a group of bits and saving memory space by packing bit sequences. To learn defining new types for a finite set of values as enumerations. Using the enumerations in different constructs of a program.

To present knowledge of the preprocessing stage in a C program compilation and to introduce the directives supported by the C preprocessor.

There are two ways to represent discrete values: as bits and as elements in a finite set.

Structure of the Lesson:

- 18.1. Bitwise operators expressions**
- 18.2. Enumeration types**
- 18.3. The C preprocessor**
- 18.4. The type definitions and macros in `stddef.h`**
- 18.5. Structures**
- 18.6. Unions**
- 18.7. Summary**
- 18.8. Technical terms**
- 18.9. Model questions**
- 18.10. References**

18.1. Bitwise Operators and Expressions:

Bitwise operators act on integral expressions represented as strings of binary digits. These operators are explicitly system dependent. The bitwise operators available in C language are listed in the following table.

Logical operators	(unary) bitwise complement	~	Note: The operator ~ is unary. All the other bitwise
	Bitwise and	&	
	Bitwise exclusive or	^	

	Bitwise or		operators are binary.
Shift operators	Left shift	<<	
	Right shift	>>	

Bitwise complement:

The operator `~` is called the one's complement operator or bitwise complement operator. It inverts the bit string representation of its argument. The 0's become 1's and the 1's become 0's. The following example shows the working of `~` operator.

```
int a = 70707;           // a is a 4 byte integer.
```

The binary representation of variable `a` is 00000000 00000001 00010100 00110011. The expression `~a` is the bitwise complement of `a`. This expression has the binary representation 11111111 11111110 11101011 11001100. The int value of the expression is `-70708`.

Two's complement:

Two's complement is used to represent negative integers. For a nonnegative integer `n`, its bit string is obtained by writing `n` in base 2. One's complement of that bit string is obtained by using the operator `~`. By adding 1 to the one's complement we obtain two's complement representation of `-n`. For example the bit string for 7 is 00000111. Its one's complement is 11111000. By adding 1 to the one's complement using binary arithmetic we obtain 11111001. This is the two's complement representation for `-7`. (For convenience only the lower order byte has been shown.) On a two's complement machine the operation `a-b` is the same as `a+(-b)` and `-b` is represented in the two's complement, i.e., to subtract `b` from `a`, the two's complement of `b` is added to `a` using binary arithmetic.

Bitwise binary logical operators:

The bitwise logical operators (`&`, `|`, `^`) use integral expressions as operands. The following table explains the effect of these operators in bits.

A	B	A & B	A B	A ^ B
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	1	1	0

Example:- Let `int a=33333, b= -77777`

```
Bit string of a   = 00000000 00000000 10000010 00110101
Bit string of b   = 11111111 11111110 11010000 00101111
a&b               = 00000000 00000000 10000000 00100101
a|b               = 11111111 11111110 11010010 00111111
a^b               = 11111111 11111110 01010010 00011010
```

Left and Right shift operators: The bitwise shift operators move the bits in a bit string left or right. Both the shift operators are binary operators. Each operator requires two integral expressions as operands. The left shift operator is applied using the following syntax.

```
expr1 << expr2
```

This operator causes the bit representation of `expr1` to be shifted to the left by the number of places specified by `expr2`. On the low-order end 0's are shifted in. Integral promotions are performed on each of the operands. For example let

```
int a=1;
```

Then the expression `a<<31` shifts 1 to the high bit(i.e. the most significant bit position).

The right shift operator is used with the syntax `expr1 >> expr2`. This operator causes the bit representation of `expr1` to be shifted to the right by the number of places specified by `expr2`. For unsigned integral expressions, 0's are shifted in at the high end. For signed types some machines shift in 0's while others shift in sign bit.

Masks: A mask is a constant or variable that is used to extract desired bits from another variable or expression. For example, the constant 255 (which is 2^8-1) has the lower order byte turned on in its bit string. So it can be used as a mask to extract the lower order byte in an integer expression, i.e., an expression `v&255` will yield integer with the bit string where higher order 3 bytes set to zero and lower order byte the same as that of `v`.

Printing an int bitwise;

Software tools are utilities that the programmer can use to write software. Most systems provide variety of software tools. The following is one such tool, which uses a mask to print out the bit representation of an `int`. The function can be used to explore how values of expressions are represented in memory.

```
/* Bit print an int expression. */
#include<limits.h>
/* CHAR_BIT is usually defined to 8 in most systems. */
void bit_print(int a)
{
    int l;
    int n= sizeof(int) * CHAR_BIT;
    int mask = 1 << (n-1);

    for(l = 1;l <= n; ++l)
    {
        putchar(((a & mask) == 0) ? '0' : '1');
        a <<= 1;
        if(i % CHAR_BIT == 0 && i< n)
            putchar(' ');
    }
}
```

Packing and unpacking:

The use of bitwise expressions allows for data compression across byte boundaries. This is useful in saving space and time. The following function can be used to pack four characters into an **int**. it uses shift operations to do the packing byte by byte.

```
In file pack_bits.c
/* pack 4 characters into an int. */
#include<limits.h>
    /* CHAR_BIT is usually defined to 8 in most systems. */
int pack(char a, char b, char c, char d)
{
    int p = a;
    p = (p << CHAR_BIT) | b;
    p = (p << CHAR_BIT) | c;
    p = (p << CHAR_BIT) | d;
    return p;}

```

18.2. Enumeration types:

An enumeration is a finite set of elements. We can define enumeration types in C language using '**enum**' keyword with the following syntax.

```
enum type-name { value1, value2, value3, ....., value n};
```

An example for an enumerated type is

```
enum day { sun, mon, tue, wed, thu, fri, sat};
```

In this example, day is the user defined enumerated type name. Its values are the identifiers from sun to sat. These values are defined as constants of type **int**. By default the first constant is 0 and each succeeding one has the next integer value. However the enumerators can be initialized with other **int** type constants. For example:

```
enum suit { clubs=1, diamonds, hearts, spades};
```

In this example the enumerators' diamonds, hearts, spades get the constants 2,3, & 4 respectively.

18.3. The C Preprocessor:

The preprocessor enhances the power of C language. It allows the programmer to take compile time decisions, to link a number of modules and to simplify the code by using macros. C language supports the following directives to the preprocessor.

```
#include    #define    #if    #ifdef    #endif    #ifndef    #elif    #else
            #undef
```

Preprocessor directives begin with `#`. The effect of a preprocessor directive starts at its place and continues until end of that file or until its effect is negated by using another directive.

#include directive: It is used to include the contents of another file into the present file. When this directive is found the preprocessor replaces the line with a copy of the contents of the named file. It searches for the file in the current directory and then in other system dependent directories. The syntax for using this directive is as follows:

```
#include<filename> or #include "filename"
```

Example:- `#include<stdio.h>` This line includes the header file `stdio.h` into the present file.

#define directive:- This directive is used to define symbolic constants and macros. It occurs in two forms. They are

```
#define identifier token_string  
#define identifier (identifier, identifier) token_string
```

In the above forms `token_string` is optional. A long definition can be continued to the next line by placing a backslash `\` at the end of the current line.

Examples: `#define PI 3.14159`
`#define sum(a,b) ((a)+(b))`

The use of `#define` improves program clarity and portability. When the preprocessor finds the first form of this directive it replaces every occurrence of `identifier` by `token_string` in the remainder of the file except in quoted strings.

The second form of this directive is also called 'macro'. The first identifier in this form is the macro name and the identifiers in parentheses are arguments for the macro. A macro may contain zero or more arguments. When such macros are called from a program, every occurrence of the macro is expanded by the preprocessor, as given in its definition. For example the macro call `sum(c, d/f)` is expanded to `(c + (d/f))`.

18.4. The type definitions and macros in `stddef.h`:

C provides the **typedef** facility to associate an identifier with a specific type.

For example:

```
typedef char uppercase;
```

makes `uppercase` a type that is synonymous with `char` and it can be used in declarations.

The header file `stddef.h` contains some implementation specific type definitions and macros. They are listed below.

```
typedef          int ptrdiff_t;  
typedef short   wchar_t;  
typedef unsigned size_t;
```

```
#define NULL ((void*) 0)
```

The Macros in `stdio.h` and `ctype.h`:

The C system provides macros `getc()` and `putc()` in `stdio.h`. `getc()` is used to read a character from a file. `putc()` is used to write a character to a file. `stdio.h` also contains the following macros.

```
#define getchar() getc(stdin)
#define putchar( c ) putc((c), stdout)
```

The C system also provides some macros in the standard header file **`ctype.h`**. These macros are used to test characters and conversion of characters. Some of the macros from **`ctype.h`** are listed below.

Macro	Nonzero (true) is returned if:
<code>isalpha(c)</code>	c is a letter
<code>isupper(c)</code>	c is an uppercase letter
<code>islower(c)</code>	c is a lowercase letter
<code>isdigit(c)</code>	c is a digit
<code>isalnum(c)</code>	c is a letter or a digit

Conditional compilation:

The preprocessor has directives for conditional compilation. They are:

`#if`, `#ifdef`, `#ifndef`, `#endif`, `#undef`. The syntax of using these directives is as below.

```
#if const_integral_expression
#ifdef identifier
#ifndef identifier
#endif
#undef identifier
```

The three directives `#if`, `#ifdef`, `#ifndef` provide for conditional compilation of the code that follows, until `#endif` is reached. For the intervening code to be compiled, after `#if` the constant expression must be non-zero, and after `#ifdef` the named identifier must be defined previously in a `#define` directive and not undefined using `#undef` until the current line. After `#ifndef`, the named identifier must be currently undefined. The integral constant expression used in a preprocessing directive cannot contain the **`sizeof`** operator or a cast. It may use the defined preprocessing operator. This operator is available in ANSI C but not necessarily in traditional C. The expression:

`defined identifier` is equivalent to `defined(identifier)`

Examples:

The following lines include the contents of the file "myfile.h", into the present file only if the symbolic constant 'FILE__H' is currently undefined.

```
#ifndef FILE__H
```



```
#include "myfile.h"
#endif
```

The predefined Macros:

In ANSI C there are five predefined macros. They cannot be undefined by the programmer. The following table lists the predefined macros and their values.

Predefined macro	Value
<code>__DATE__</code>	A string containing the current date
<code>__FILE__</code>	A string containing the file name
<code>__LINE__</code>	An integer representing the current line number
<code>__STDC__</code>	If the implementation follows ANSI standard C, then the value is a nonzero integer.
<code>__TIME__</code>	A string containing the current time.

The operators # and ##:

The preprocessing operators # and ## are available in ANSI C. The unary operator # causes 'stringization' of a formal parameter in a macro definition. The binary operator ## is used to merge tokens.

Examples:

```
#define message_for(a,b) \
    printf("#a," and "#b ": We love you!\n")
int main(void)
{
    message_for(Carole,Debra);
    return 0;
}
```

The macro call in the above example will be replaced by `printf("Carole" " and " "Debra" ": We love you!\n");` this is equivalent to `printf("Carole and Debra We love you!\n");`

Example 2:-

```
#define X(i) x ## i
X(1) = X(2) = X(3);
```

The above line in the example 2 will be replaced by `x1 =x2=x3;`

The assert() Macro:

This macro is defined in `assert.h` header file. It is used to ensure the value of an expression is what you expect it to be.

Example:

```
char * p;
```

```
-----
```

```
-----  
assert(p != NULL);
```

The above `assert()` macro compares the pointer variable `p` to `NULL`. If it is equal to `NULL` it will abort the program.

#error and #pragma:

ANSI C added the `#error` and `#pragma` preprocessing directives. `#error` is used to output the compile-time errors. `#pragma` directive is provided for implementation-specific uses. The general forms of these directives are

```
#error "Compile-time error message."  
#pragma tokens
```

18.5. Structures:

A structure is a data type, which combines a variety of pieces of data together into one type.

A structure is formed in C by using the keyword "struct" followed by an optional tag-field and then a list of members within the structure.

The syntax is:

```
struct structure-tag-field  
{  
    member-type member variable1;  
    member-type member variable2;  
} structure-variable-list;
```

The structure definition terminates with a semicolon(;) because it is a C language statement. The tag-field and structure-variable-list are optional. But at least one of the two should be specified.

```
For Ex:    struct student  
            {  
                int sno;  
                char sname[10];  
                float per;  
            };
```

Here `student` is tag-field name. The tag-field name helps the programmer to create any number of structure variables of that type.

Accessing member:

For accessing members of the structure variable, you can use the member selector operator(`.`) or (`→`). While accessing through a variable you can use the (`.`) operator. While accessing through pointer you can use the `→` operator.

Initializing structures:

Initializing an object implies “providing a value to a variable at creation”. We can initialize a structure variable at declaration itself.

Ex: Struct student

```
{
    int sno;
    char sname[10];
    float per;
    }s={1,"ramu",82.05}
```

write a program to display time by using structures.

```
# include<stdio.h>
struct time
{
    int hur;
    int minute;
    int second;
} now;
main()
{
    struct time later;
    now.hour = 10;
    now.minute = 30;
    now.second = 4;
    later = now;
    printf("time is %d:%2d:%2d\n", later.hour, later.minute, later.second);
}
```

This declares structure tag, **struct** time, which keeps the members, hour, minute, second, in one piece. The variables now and later are structures of type **struct** time, declared the same way as integers. The members of the structure can be any type: int, float, double, char, arrays, other structures, and pointers to any of these. To access the members, there are two operators available (. and ->). To access the specified member, use the dot(.) operator.

Valid structure operations:

- Assigning to or copying it as a unit
- Accessing its members
- Taking its address with “&”
- Passing it as an argument to functions, returning it by functions
- Initializing it with a list of constant member values, or by assignment for automatic structures
- Initialization is not allowed with "Traditional C," so a separate assignment statement should be used.

Constructing an array of structures:

An array is a collection of objects of same type. Each element in the array is identified by index.

```
struct studet
{
    int sno;
    char sname[10];
    int m1,m2,m3;
```

```
}s[10];
```

it declares an array of student variable with the name 's'. If you want to access those variables, each structure variable requires an index. Index starts with 0.

Ex: #include<stdio.h>

```
#include<conio.h>
```

```
struct emp
```

```
{
    int eno;
    char ename[10];
    int sal;
```

```
}e[10];
```

```
void main()
```

```
{
```

```
    int l,n;
```

```
    printf("enter the value of n:\n");
```

```
    scanf("%d",&n);
```

```
    for(l=0;l<=n;l++)
```

```
    {
```

```
        printf("enter emp information: \n");
```

```
        printf("enter eno: \n");
```

```
        scanf("%d",&e[l].eno);
```

```
        printf("enter ename:\n");
```

```
        scanf("%s",e[l].ename);
```

```
        printf("enter sal: \n");
```

```
        scanf("%d",& e[l].sal);
```

```
    }
```

```
    printf("emp information:\n");
```

```
    for(l=0;l<=n;l++)
```

```
    {
```

```
        printf("%d %s %d",e[l].eno,e[l].ename,e[l].sal);
```

```
    }
```

```
    getch();
```

```
}
```

Pointers to structures:

A pointer is a memory block, which contains the address of an object. A pointer to structures can be declared by using the tag_name of the structure.

Ex:

```
struct student
{
    int sno;
    char sname[10];
    char address[20];
};
student *s;
```

The above declaration declares a pointer to the structure variable of type "student". A valid address should be supplied to the pointer variable so that its access is legal. To access the member of the structure variable we should use '->' operator.

Passing structures to functions:

A function is a block of code that performs a specific task. Sometimes a function requires information in order to process the block, that information will be supplied to the function in the form of arguments.

To receive those arguments you declare parameters in the function header. A variable can be passed to the function directly (call by value) or an address of the variable can be send (call by reference).

Passing array of structures to functions:

Array always passes to the functions through call by reference. The array of structures should also be passed to the function using call by reference only.

18.6. Unions:

Unions are class type that can contain only one data element at a time

The member of a union represents the kinds of data that the union represents and the kinds of data that the union can contain. An object of union type requires enough storage to hold the largest member in its member list.

A union is constructed by using the key word 'union'.

Syntax:

```
Union tag field
{
    datatype member variable 1;
    datatype member variable 2;
    -----
    datatype member variable n;
};
```

A semicolon is used for termination because the structure definition is a 'C' statement.

Ex: Union myunion

```
{
    char c;
```

```
    int ival;
    float fvalue;
    double dvalue;
}u;
```

The union is defined with the key word 'union' followed by an optional tag field i.e., u. This union contains several members : a character, an integer, a float and a double.

18.7. Summary:

The bitwise operators give programmer the means of accessing the bits in integral expressions. The use of bitwise expressions allow for data compression across data boundaries. This compression is useful in saving space and time.

Bitwise operators are machine-dependent. Masks are particular values used typically with the '&' operator to extract a given series of bits. Packing is the art of placing a number of distinct values into various subfields of a given variable. Unpacking extracts these values.

The keyword **enum** allows the programmer to define enumeration types. A variable of such a type can take values from the set of enumerators associated with the type.

Enumerators are distinct identifiers chosen for their mnemonic significance. Enumerators are constants of type **int**. Thus they can be used in case labels in a **switch**.

The preprocessor provides facilities to include other files into a program file. It allows defining macros for long expressions. Preprocessor provides general macro facility with argument substitution. The assert () macro can be used to ensure that an expression has an appropriate value.

In this lesson, we have described the last remaining data types provided by the C language: structures and unions. A structure allows the grouping of various pieces of related information of different types into one variable.

Finally, we have described the union data type, which is defined similar to structures; however, the semantics of only one of the member types being resident in such a variable at one time. That is, a union allows several different types of information to be stored in the same physical space at different times. For a union variable, storage is allocated only for the largest of the data types which may reside in the variable.

Structures are a valuable tool for developing complex programs and data structures in an efficient and top down manner.

18.8. Technical terms:

Binary: A numbering system with only two values: 0 (zero) and 1 (one).

Address: Every variable, or object, within a program is located at a particular address in memory. In C/C++, addresses may be assigned to pointers.

Preprocessor: A language processor that is used to translate and perform other types of manipulations of a program before it is compiled or assembled.

Macro: A macro is an abbreviation for a set of commands.

Compilation: Compilation is the process of creating an executable program from source code.

18.9. Model Questions:

1. Explain how masks are useful in extracting bits?
2. Define enumeration type? Give an example?
3. What is a structure? Explain how to create array of structures?
4. Differentiate structure and union with an example?
5. Explain the working of any five preprocessor directives?

18.10 References:

Byron C. Gottfried “**Programming with C**”

Kelly & Pohl “**A book on C**”

Yeshavanth kanethkar “**Let us C**”

Ravi Chandran “**Programming in C**”

Yeshavanth kanethkar “**Working with C**”

Blaguruswamy “**ANSI C**”

-Y. VENKATESWARA RAO, M.C.A.,

Lecturer, Dept.Of Computer Science,

JKC College, GUNTUR.

Lesson : 19**LIST PROCESSING AND I/O****19.0 Objectives:**

To present the concept of grouping elementary items through structures and unions, to get higher-level data types. To create linear and non-linear data structures like Linked Lists and Trees. To appreciate the application of self-referential structures in building linked data structures.

Input/Output is the most common requirement for any program. Every open system depends on Input/ Output activities to interact with its users. This chapter presents a description of input/output functions in C language. I/O functions that are used with files, and strings have also been described. The UNIX operating system is known for its security aspects. This chapter presents a brief discussion of the file access permissions in UNIX operating system.

Structure of the Lesson:

- 19.1. Structures and List Processing**
- 19.2. List Operations**
- 19.3. Stacks**
- 19.4. Queues**
- 19.5. Binary Trees**
- 19.6. Binary Tree Traversal**
- 19.7. Input Output and Operating System**
- 19.8. Summary**
- 19.9. Technical terms**
- 19.10. Model Questions**
- 19.11. References**

19.1.Structures and List Processing:**Self-Referential Structures:**

A structure may contain pointer variables as its members. One or more of these member variables may be pointers to parent structure. Such a structure is called self-referential structure. In general terms this can be expressed as:


```

struct tag{
    member 1;
    member 2
    -----
    struct tag *name;
};

```

In the above syntax name refers to the name of a pointer variable. Thus structure of type *tag* will contain a member that points to another structure of type *tag*.

Self-referential structures are useful in building linked data structures such as lists and trees. An example of a self-referential structure is given below.

```

struct node
{
    int data;
    struct node *next;
};

```

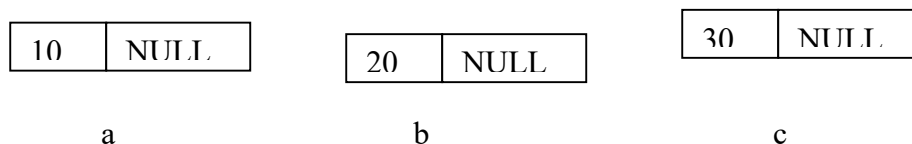
The member 'next' in the above structure can be used to link each node type variable to another node. The following example depicts this linking.

```

struct node a,b,c;
a.data = 10;
b.data = 20;
c.data = 30;

```

Before linking the nodes a,b,c exist as ...

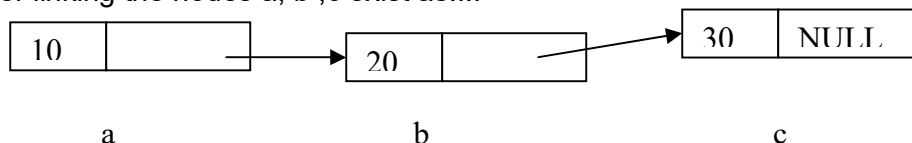


```

a.next=&b;
b.next=&c;

```

After linking the nodes a, b ,c exist as....



Such a collection of linked nodes is usually called linked list or nodes.

Linear Linked List:

There are several kinds of linked data structures. One among them is the linear linked list. In this data structure all the nodes in the list are linked linearly in sequential order.

Storage allocation:

One major advantage of a linear linked list is that it can be of varying length. The number of nodes in the list can be dynamically decided, as each node can be dynamically allotted storage using **malloc()** function in the standard library. This function is called with the syntax:

```
malloc(size);
```

It returns a pointer to enough storage for an object of 'size' number of bytes. To dynamically create a node, this function is called with the '**sizeof(node)**' as argument. For example the following lines dynamically create a node.

```
struct node *head;  
head = malloc(sizeof(node));
```

19.2.List Operations:

The basic operations on a linear linked list are listed below.

1. Creating a list.
2. Counting the elements in a list.
3. Looking up an element.
4. Concatenating two lists.
5. Inserting an element into the list.
6. Deleting an element from the list.

The list operations can be defined as functions. For these functions the following *list.h* header file is required.

```
/* list.h header file. */  
#include<stdio.h>  
#include<stdlib.h>  
typedef char DATA;  
struct node  
{  
    DATA d;  
    struct DATA *next;  
};  
typedef struct node ELEMENT;  
typedef ELEMENT* LINK;
```

Creating a list:

The following function will produce a list of 'nodes' as defined in the list.h from a string. The function will return a pointer to the head of the resulting list.

```
/* List creating function using recursion. */  
#include "list.h"
```

```
LINK string_to_list(char s[])
{
    LINK head;
    if(s[0] == '\0')
        return NULL;
    else{
        head = malloc(sizeof(ELEMENT));
        head->d = s[0];
        head->next = string_to_list(s+1);
        return head;
    }
}
```

The `string_to_list()` function takes a string as the argument. It is a recursive function. For this function the base case (i.e., the case that terminates recursive calls) is `s[0] == '\0'`, i.e., when the end of the string is detected. In every recursive call the member 'd' of the allocated element is assigned the first character in the string `s[]`. The pointer expression `s+1` points to the remainder of the string. The function is called recursively with `s+1` as argument. Finally the function exits with the address of the head of the list.

```
/* List creating using iteration. */
```

```
#include "list.h"
LINK s_to_l(char s[])
{
    LINK head = NULL, tail;
    int i;
    if(s[0] != '\0')
    {
        head = malloc (sizeof(ELEMENT));
        head->d = s[0];
        tail = head;
        for(i=1; s[i] != '\0'; ++i)
        {
            tail->next = malloc(sizeof(ELEMENT));
            tail = tail->next;
            tail->d = s[i];
        }
        tail->next = NULL;
    }
    return head;
}
```

This function creates a linear linked list of nodes from a string using iteration. If a null is given the `s_to_l()` function it produces an empty list by returning `NULL`.

Counting the nodes in a list:

The following function counts the number of nodes in a list. The function `count()` does this recursively where as `count_it()` does this iteratively. These two traverse the list until the 'next' member of a node is NULL.

```
/* count a list recursively. */
```

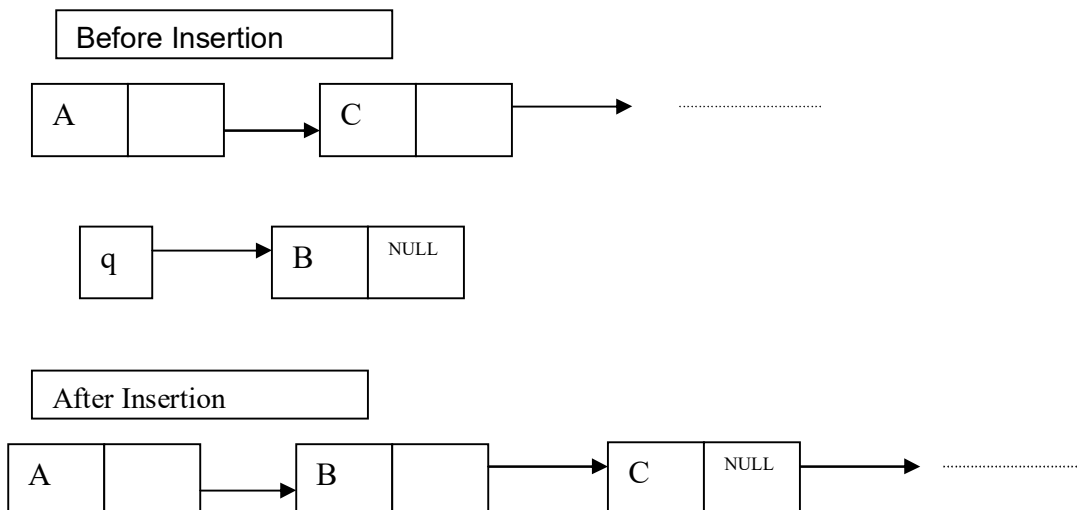
```
int count(LINK head)
{
  if(head == NULL)
    return 0;
  else
    return(1+count(head->next));
}
```

```
/* Count a list iteratively. */
```

```
int count_it(LINK head)
{
  int cnt=0;
  for(; head!=NULL; head= head->next)
    ++cnt;
  return cnt;
}
```

Inserting a node into a list:

One of the most useful properties of a list is that insertion takes a fixed amount of time, once the position in the list is found. The following picture illustrates insertion of a node into a list.



The following function inserts a node into a linear list.

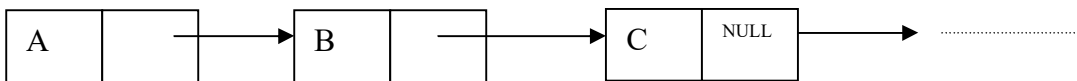
```
void insert(LINK p1, LINK p2, LINK q)
{
    assert(p1->next == p2);
    p1->next = q;
    q->next = p2;
}
```

The insert() function places the element pointed at by q between the elements pointed to by p1 and p2.

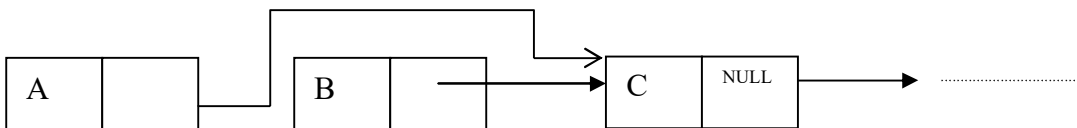
Deleting a node from a linear list:

To delete a node from a list, its predecessor node is adjusted such that its 'next' member is assigned the successor to the deleted element. The following picture illustrates the deletion of a node from a list.

Before Insertion



After Insertion



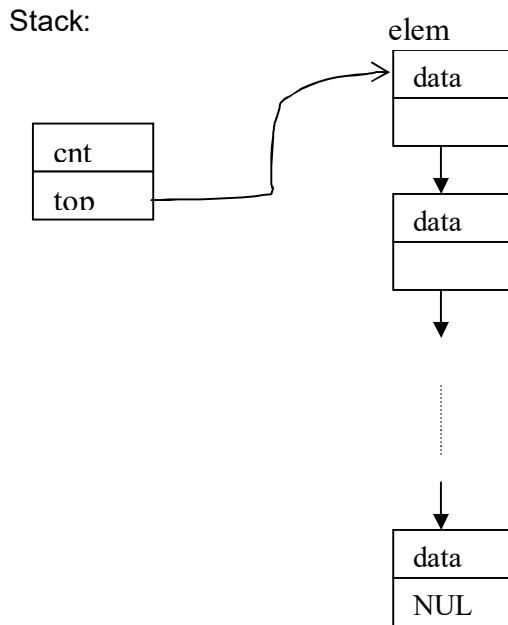
The following function deletes all the nodes in a list and frees the memory occupied by the nodes. It recursively traverses the list and deletes each node.

```
void delete_list(LINK head)
{
    if(head != NULL)
    {
        delete_list(head->next);
        free(head);
    }
}
```

19.3. Stacks:

The stack is a data structure that allows insertion and deletion of data to occur only at a single restricted element, the top of the stack. This is the last-in-first-out (LIFO) discipline. A stack behaves like a pile of trays. The typical operations that can be used with the stack are **push**, **pop**, **top**, **empty**, **full** and **reset**. The **push** operation places a value on the stack. The **pop** operation retrieves and deletes a value off the stack. The **top** returns the top element of the stack. The **empty** and **full** test the stack to be empty and full respectively.

A stack implementation:



The following program illustrates the operations on a stack.

```

/* A linked list implementation of a stack. */
#include<stdio.h>
#include<stdlib.h>

#define EMPTY 0
#define FULL 10000

typedef char DATA;
typedef enum {false, true} boolean;

struct elem{
    DATA d;
    struct elem *next;
  
```

```
};

typedef struct elem elem;

struct stack_t{
    int cnt;
    elem *top;
};

typedef struct stack_t stack;

void initialize(stack *stk);
void push(data d, stack *stk);
DATA pop(stack *stk);
DATA top(stack *stk);
boolean empty(const stack *stk);
boolean full(const stack *stk);

/* basic stack routines. */

void initialize(stack *stk)
{
    stk->cnt = 0;
    stk->top = NULL;
}

void push(DATA d, stack *stk)
{
    elem *p;

    p = malloc(sizeof(elem));
    p->d = d;
    p->next = stk->top;
    stk->cnt++;
}

void pop(stack *stk)
{
    DATA d;
    elem *p;

    d = stk->top->d;
    p = stk->top;
    stk->top = stk->top->next;
    stk->cnt--;
    free(p);
    return d;
}
```

```
}

DATA top(stack *stk)
{
    return (stk->top->d);
}

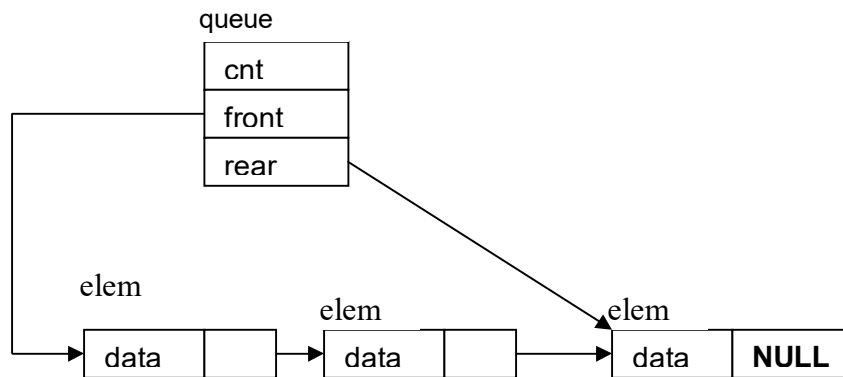
boolean empty(const stack *stk)
{
    return ((boolean) (stk->cnt == EMPTY));
}

boolean full(const stack *stk){
    return ((boolean) (stk->cnt == FULL));
}

/* Test the stack implementation by reversing a string. */
int main(void)
{
    char str[] = "Acharya Nagarjuna University, Guntur.";
    int i;
    stack s;
    initialize(&s);
    printf(" In the string : %s\n",str);
    for(i=0; str[i] != '\0'; ++i)
        if(!full(&s))
            push(str[i],&s);
    printf("From the stack: ");
    while(!empty(&s))
        putchar(pop(&s));
    putchar('\n');
    return 0;
}
```

19.4. Queues:

A **queue** is an abstract data type that can be implemented using a linear list. A queue is a FIFO structure with two ends, which are called the rear end, and the front end. Insertion occurs at the rear of the list, and deletion occurs at the front of the list.

A queue implementation:**19.5. Binary Trees:**

A tree is a finite set of elements called nodes. A tree has a unique node, called the root node, where the remaining nodes are a disjoint collection of subtrees. If node r has T_1, T_2, \dots, T_n as subtrees, then r_1, r_2, \dots, r_n , the roots of these subtrees, are the children of r . A node with no children is called a leaf node.

A binary tree is a tree whose elements have two children. A binary tree considered as a data structure is an object made up of elements that are characterized by two link fields, called left child and right child. Each link must point at a new object not already pointed at or be NULL. A leaf node has both left and right child as the value NULL. The following structures and type specification will be used to define binary trees.

```
/* file tree.h */
```

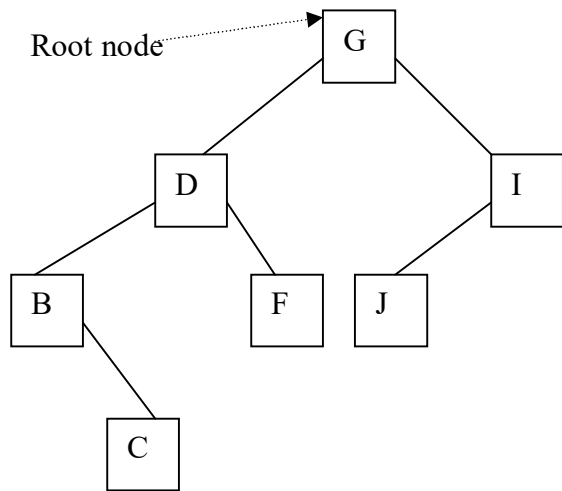
```
#include<assert.h>
#include<stdio.h>
#include<stdlib.h>
```

```
typedef char DATA;
```

```
struct node{
    DATA D;
    struct node *left, *right;
};
```

```
typedef struct node NODE;
typedef NODE *BTREE;
```

```
#include "fct_proto.h"
```

An example for a Binary Tree:**19.6. Binary Tree Traversal:**

There are several natural ways to visit the elements of a binary tree. The three commonly used traversals are **inorder** traversal, **preorder** traversal, **postorder** traversal. These traversal methods are defined using recursive algorithms. The algorithms and the implementation in C language are given below.

Preorder Traversal:

Step 1: Visit the root node.

Step 2: Traverse the left subtree in preorder.

Step 3: Traverse the right subtree in preorder.

Inorder Traversal:

Step 1: Traverse the left subtree in inorder.

Step 2: Visit the root node.

Step 3: Traverse the right subtree in inorder.

Postorder Traversal:

Step 1: Traverse the left subtree in postorder.

Step 2: Traverse the right subtree in postorder.

Step 3: Visit the root node.

```
/* Implementation for Preorder traversal of a binary tree. */
```

```
void preorder(BTREE root)
```

```
{  
  if(root != NULL)  
  {  
    printf("%c",root->d);  
    preorder(root->left);
```

```

    preorder(root->right);
}
}

/* Implementation for Inorder binary tree traversal. */
void inorder(BTREE root)
{
    if(root != NULL)
    {
        inorder(root->left);
        printf("%c",root->data);
        inorder(root->right);
    }
}

/* Implementation for Postorder traversal of a binary tree. */
void postorder(BTREE root){
    if( root != NULL){
        postorder(root->left);
        postorder(root->right);
        printf("%c",root->d);
    }
}

```

19.7. INPUT / OUTPUT And Operating System:

The output function printf():

The **printf()** function delivers its character stream to the standard output file “**stdout**”, which is normally connected to the screen. The argument list to **printf()** has two parts:

control_string and other_arguments.

The expressions in other_arguments are converted according to the formats in the control string and then placed in the output stream. Characters in the control string that are not part of a format are placed directly in the output stream. The ‘%’ symbol introduces a conversion specification or format. A single conversion specification begins with % and ends with a conversion character.

printf() conversion characters.	
Conversion character	How the corresponding argument is printed.
c	As a character
d, i	As a decimal integer
u	As an unsigned decimal integer
o	As an unsigned octal integer

X, x	As an unsigned hexadecimal integer
e	As a floating-point number (scientific notation)
E	As a floating-point number (scientific notation with E)
f	As a floating-point number (fixed point notation)
g	In the e-format or f-format which ever is shorter
G	In the E-format or f-format which ever is shorter
s	As a string.

The conversion specification may also contain the following information:

- > Zero or more flag characters
- > An optional positive integer to specify width of the field
- > An optional precision number.

The effect of the above given parts in the conversion specification are as stated below.

Flag characters:

These flags modify the meaning of the conversion specification. The flags are as listed below:

Flag Symbol	Meaning
-	A minus sign left adjusts the converted argument.
+	A plus sign prepends the symbol + to non-negative numbers coming from signed conversion.
#	Means that the result be converted to an alternative form depending on the conversion character.
0	Zeros are used to pad the field instead of blanks.

The width conversion specification:

An optional positive integer may be used to specify minimum field width of the converted argument. If the converted argument has fewer characters than the specified field width, then it will be padded with spaces on the left or right. If the converted argument has more characters than the specified field width, then the field width will be extended to the required size.

The precision:

An optional precision number may be given as a period followed by a non-negative integer. For d, i, o, u, x, or X conversions, it specifies the minimum number of digits to be printed. For e, E and f conversions, it specifies the number of digits to the right of the decimal point.

Example for printf():

```
printf("she sells %d %s for f", 99, "sea shells", 3.77);
```

The input function scanf():

The function **scanf()** reads characters from the standard input file "**stdin**". The argument list to **scanf()** has two parts:

Control_string and other_arguments

scanf() conversion characters.		
Conversion character	Characters in the input stream that are matched	Corresponding argument is pointer to
c	Any character including white space.	char
d, i	Decimal integer	int
u	Unsigned decimal integer	unsigned int
o	Octal integer	unsigned int
X, x	Hexadecimal integer	unsigned int
E, e, f, g, G	Floating-point number	float
s	A sequence of characters	char

A conversion specification of the form %[string] indicates that, a special string is to be read in. If the first character in the string is not a circumflex character (^), then the string is to be made up only of the characters in the string. If the first character in the string is circumflex (^) then the string is to be made up of the characters other than those in the string. This format is particularly useful to read in paragraphs of text.

The functions fprintf(), fscanf(), sprintf(), and sscanf():

The functions **fprintf()** and **fscanf()** are file versions of the functions **printf()** and **scanf()**, respectively. The prototypes of these functions are as stated below.

```
int fprintf(FILE *fp, const char *format, ...);
int fscanf(FILE *fp, const char *format, ...);
```

The identifier FILE is defined in **stdio.h** file. It is a structure that describes the current state of a file.

The functions **sprintf()** and **sscanf()** are string versions of the functions **printf()** and **scanf()**, respectively. Their function prototypes are as given below.

```
int sprintf(char* s, const char *format, ...);
int sscanf(const char *s, const char *format, ...);
```

The function **sprintf()** writes to its first argument, a pointer to char (string). Its remaining arguments conform to those for **printf()**. The function **sscanf()** reads from its first argument instead of from keyboard. Its remaining arguments conform to those for **scanf()**.

The functions **fopen()** and **fclose()**:

Files have several important properties. Every file has a name. A file must be opened and closed. They can be written to, or read from, or appended to. Conceptually until a file is opened nothing can be written to it or read from it. When you finish reading and writing operations on a file, you should close the file. The functions **fopen()** and **fclose()** are used to open a file and close an opened file. The syntax for **fopen()** function call is:

```
fopen(filename, mode)
```

The filename in the above syntax refers to the name of the file to be opened and mode refers to the mode of operating the file. Mode can be one of the following in the list.

Mode	Meaning
"r"	Open text file for reading.
"w"	Open text file for writing.
"a"	Open text file for appending.
"rb"	Open binary file for reading.
"wb"	Open binary file for writing.
"ab"	Open binary file for appending.

Accessing a file randomly:

The library functions **fseek()** and **ftell()** are used to access a file randomly. An expression of the form **ftell(file_ptr)** returns the current value of the file position indicator. The value represents the number of bytes from the beginning of the file counting from zero. The function **fseek()** takes three arguments. They are:

A file pointer, an integer offset, and an integer that indicates the place in the file from which the offset should be computed.

A statement of the form `fseek (file_ptr, offset, place)` sets the file position indicator to a value that represents *offset* bytes from *place*. The functions `fseek()` and `ftell()` will work properly on binary files only.

File access permissions:

In UNIX a file is created with associated access permissions. The permissions determine access to the file for the owner, the group, and for others. The access can be read, write, execute, or any combination of these, including none. File permissions can be set using octal notation or mnemonics. The meanings of octal digits in the file permissions are listed below.

Meaning of octal digits in file permissions		
Mnemonic	Bit representation	Octal representation
r--	100	04
-w-	010	02
--x	001	01
rw-	110	06
r-x	101	05
-wx	011	03
rx	111	07

Executing commands from a C program:

The library function `system()` provides access to operating system command. The syntax of this function is `system(string)`. The string passed to the `system()` function is treated as an operating system command. When this function is executed the control is passed to the operating system, the command is executed, and then control is passed back to the program.

Example: `system("date");` // prints the system date.

19.8. Summary:

Self-referential structures use pointers to address identically specified elements. The function `malloc()` is used to dynamically allocate storage. The function `free()` is a storage management routine that returns unused storage.

The abstract data type (ADT) stack can be implemented using arrays and linear linked lists as well. It allows operations like `push()`, and `pop()`.

The ADT queue is a FIFO data structure. Like stack it can also be implemented using a linear linked list or an array. The basic operations allowed on a queue data structure are **insert()** and **delete()**.

Binary Trees are represented as structures with two link members. Binary trees are useful in organizing non-linear data. Binary Trees are traversed in using three common ways called preorder, inorder and postorder.

The **printf()** and **scanf()** are the most widely used I/O functions in C language. The **printf()** writes character sequences to the screen. The **scanf()** reads values from keyboard. The variants of these two functions that apply on files are **fprintf()** and **fscanf()**. Another two variants that work with strings are **sprintf()** and **sscanf()**.

19.9. Technical Terms:

Flag character: A character (or set of characters) that uniquely defines the specific use of the data.

Data structure: A data structure is a way of storing data in a computer so that it can be used efficiently.

Command : A specific instruction to a computer program, issued by the user to perform a desired action.

Linked list: A linked list is one of the fundamental data structures used in computer programming. It consists of a sequence of nodes, each containing arbitrary data fields and one or two references ("links") pointing to the next and/or previous nodes.

Tree Traversal: This is a systematic way of ordering the nodes of a tree. There are three popular schemes (many other schemes are possible): 1. Preorder 2. Inorder 3. post order.

19.10. Model Questions:

1. What a linked list? Explain various operations on list?
2. Write about stacks and queues?
3. What are various input and output operations?
4. What is a binary tree? Explain various traversal methods?

19.11. References:

Byron C. Gottfried "**Programming with C**"
Kelly & Pohl "**A book on C**"
Yeshavanth kanethkar "**Let us C**"
Ravi Chandran "**Programming in C**"
Yeshavanth kanethkar "**Working with C**"
Blaguruswamy "**ANSI C**"

-Y. VENKATESWARA RAO, M.C.A.,
Lecturer, Dept.Of Computer Science,
JKC College,GUNTUR