2nd Electives
Computer Application
(Optional)

SASTRI, B.A I YEAR
Part II - Paper - I

INFORMATION TECHNOLOGY
&
PROGRAMMING WITH 'C'

तमसो मा ज्योतिर्गमय

राष्ट्रियसंस्कृतविद्यापीठम्, तिरुपतिः
(मानितविश्वविद्यालयः)

Directorate of Distance Education

National Sanskrit University
(Erstwhile Rashtriya Sanskrit Vidyapeetha, Tirupati)
Accredited at 'A' level by NAAC
Tirupati - 517 507 (A.P.)

UNIT – I

INTRODUCTION TO COMPUTERS

Structure

1.0 Objectives

The main objective of this unit is to make clear about basic functionalities of Computer system, list input and output devices, memory devices and wisely choose the one depending one own requirement, classification of computers and identify each computer peripherals.

1.1 Introduction to computers:

People have always needed to count. Earlier they counted with their fingers and probably through scratch making on trees or bones or on the wall of their caves. Over the centuries the science of numbers and computing has changed dramatically. People are now able to solve very sophisticated problems and increasingly so with the aid of computers.

Today we use computers in virtually every environment, even in areas unconnected with number manipulation. From the executive in an office to the housewife and her washing machine to the teenager recording his/her favorite television programme, computers have entered into practically every sphere of human activity.

The present topic is introduced for traditional Sanskrit students at under graduated level for exposing them to Computer fundamentals. So that students can study the developments in the area of Computer Science and adapt them to their requirement in the relevant applications.

1.2 What is a Computer?

A computer is often understood to be device that " computes" or calculates numbers. A computer however, does more that that. It can edit text, generate pictures or graphs, translate languages and even play games.

A computer can be defined as an electronic device with the ability to accept data and instructions from user, store them and execute the instructions, output the results according to user specifications and communicate or send data images and sound through telephone or cables.

1.2.1 Why use a computer?

An organization invariable generates large volume of data. Processing this data tends to be a very tedious and time consuming task. Computers are ideally suited for repetitive tasks. They are accurate, versatile and can handle many different applications, by computerizing repetitive tasks, an organization can release personnel for more productive tasks and ensure accurate and quick processing data.

Computers are now being used in almost every sphere of our lives form complex scientific applications to the maintenance of household accounts. The presence of computer is felt in all walks of life.

- Forecasting the weather
- Processing information like student's marks

- Reservation for Railways, Airlines etc.,
- Producing music and movies for entertainment

1.2.2 Characteristics of computers

All computers have certain common characteristics irrespective of their type and size. Computers are not just adding machines. They are capable of doing complex activities and operations. They can be programmed to do complex, tedious and monotonous tasks. Computers are what they are because of the following characteristics:

- Word length
- Speed
- Storage
- Accuracy
- Versatility
- Automation
- Diligence (thoroughness)

a) **Word length:** A digital computer operates on binary numbers 0 and 1. It can understand information only in terms of 0's and 1's. A binary digit is called a **bit**. A grout of 8 bits is called a **byte.** The number of bits that a computer can process at a time in parallel is called its **word length.** Commonly used word lengths are 8, 16, 32, 64 or 128 bits. Word length is the measure of the computing power of a computer. The longer the word length, the more powerful the computer is.

b) **Speed:** Computers can calculate at very high speed. Microcomputers, for example, can executer millions of instructions per second over and over again without any mistake. As the power of the computers increases, the speed also increases. For example Supercomputers can operate at speed measured in nanoseconds and even in pico seconds - one thousand to one million times faster than microcomputer.

c) **Storage:**The capability of storing and retrieving huge amounts of data in a fast and efficient manner is one of the important characteristics of computers.

d) **Accuracy:** the accuracy of a computer system is very high. Errors in hardware can occur, but error detecting and correcting techniques will prevent false results. In most cases the errors are due to the human factors rather than the technological flaws.

e) **Versatility:** Computers are very versatile machines. They are capable of performing almost any task, provided the task can be reduced to a series of logical steps. Computers can communicate with other computers and can receive and send data in various forms like text, sound, video, graphics, etc. this ability of computer to communicate to another has led to the development of computer networks, internet, WWW and so on.

f) **Automation:** the level of automation achieved in a computer is phenomenal. It is not a simple calculator where one has to punch in the numbers and press the ‘equal to’ sign to get result. Once a task is initiated, computers can proceed on its own till its completion. Computers can be programmed to perform a series of complex tasks involving multiple programs. Computers will perform these things flawlessly. They will executer the programs in the correct sequence, they will switch on/off the machines at the appropriate time, they will monitor the operational parameters, and they will send warning signals to take corrective actions if the parameters exceed the control level, and so on. Computers are capable of these levels of automation, provided they are programmed correctly.

g) **Diligence (thoroughness):** Diligence means being constant and earnest in efforts and application. Human beings suffer from wakens like tiredness, lack of concentration etc. human have feelings, they become sad, depressed, bored, and negligent and it will reflect on the work they do. Moreover human begins cannot perform the same or similar tasks over and over again with the same precision, accuracy, and enthusiasm as the first time. After some time people will become bored and tedium will set in. this will affect the performance. Being a machine computer does not have any these human weaknesses. They won’t get tired or bored. They will not go into depression or lose concentration. They will perform the tasks given to them, irrespective of whether it is interesting creative, monotonous or boring, irrespective of whether it is the first time or the millionth time, with exactly the same accuracy and speed.

1.2.3 Anatomy of Computer System:

A computer system is a set of components that work together to accomplish one or more common goals. A computer system can be viewed as a system of three major components.

1. **Hardware:** This includes the computer equipments that is the mechanical, electrical and electronic parts of the system
2. **Software:** Programs that instruct a computer how to process the data and generate required information.
3. **Computer heart-ware**: People, who prepare data for computerized input, write computer programs, monitor computer operations and distribute the output. There are also end users who use computer resources.

A computer system functions effectively only when all three components perform their assigned tasks. The failure of any one component of the system leads to the failure of the system as a whole.

1.2.4 Logical Organization of Computer:

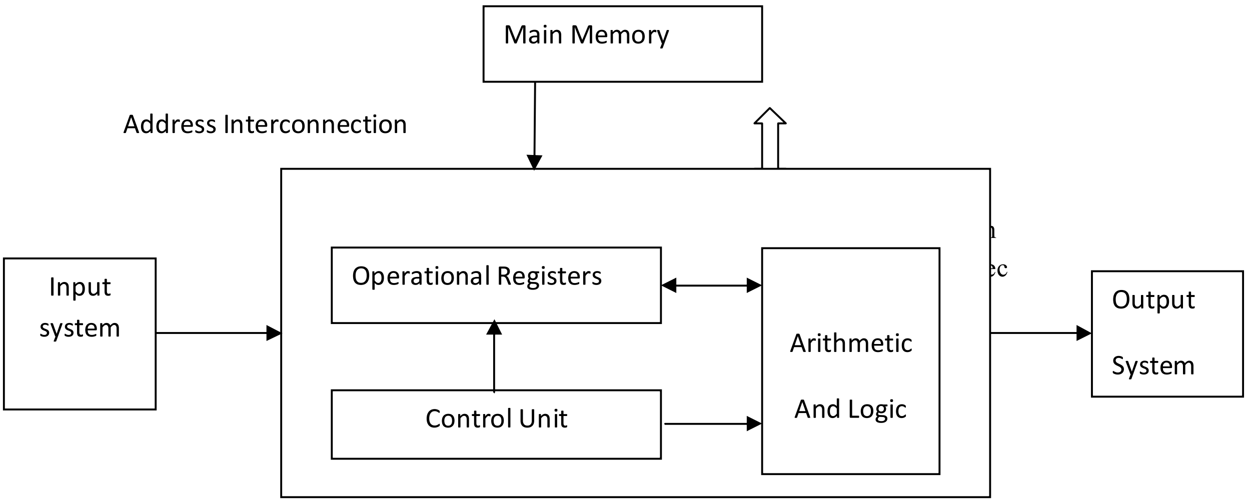


Figure 1.1 Logical organization of computer

Computer hardware falls into two categories: processing hardware, which consists of the central processing unit (CPU), and the peripheral devices. The CPU, as its name implies, is where the data processing is done. Peripheral devices allow people to interact with the CPU.

Central processing unit (CPU): The “heart” of the computer, the component that actually executes instructions. It is the CPU that understands and executes all our instructions and also coordinates the functions of other units of the computer and ensures smooth operations.

The CPU performs

a. Arithmetic operation (add, subtract, multiply, divide)
b. Logical operations (logical comparisons)
c. Input – output operations (accepting data to be processed, generating the output or results of processing)
d. Internal data movements (moving data between various parts of storage)
e. Data manipulation (accepting data, sorting it, processing it, generating the required information)

To perform these various operations the CPU has various components

1. The arithmetic logic Unit
2. The Control Unit
3. The memory Unit

ALU: The ALU performs all the arithmetic and logical operations on data selected from the memory.

Control Unit: The unit, which interprets a code to generate respective control signal, is termed as Control Unit. It interprets the instructions stored in memory, fetches the data required for the next instruction from the memory, co-ordinates the activities of ALU, the memory and other devices.

Memory Unit: This is computers work area where data and instructions gets stored before being processed and where the results get stored prior to the output. The Computer need not remember all the data, all the time. Some of the data or instructions can be kept elsewhere for retrieval at a later stage. Only the data is being processed and the instructions to process it, are stored internally where as the data and the instructions that are required later are stored externally. The internal storage is also called the primary storage, the main memory or just memory, this is usually limited. The external storage is unlimited storage.

Primary Memory:

The primary memory stores

1. The instructions waiting to be executed.
2. The instructions currently being executed.
3. The data waiting processing.
4. The data currently being processed.
5. The information awaiting output.

The control unit and internal storage are linked together by sets of parallel electrical conducting lines called buses. The buses that carry data are called data buses. To retrieve data from memory it is necessary to identify the address in main memory. The address is carried by the memory address bus.

SAQ

1. What are computer?
2. List different types of computers?
3. What ate the major components of a computer?
4. What are the characteristics of computers?
5. What do you mean by word length of a computer?
6. What do you mean by the speed of a computer?
7. What is so special about the data storage capability of a computer?
8. What do you mean by automation and diligence with respect to a computer?
9. Draw the block diagram of computer and explain?
10. What is a CPU and explain how it works?
11. What is the function of the memory?
12. Explain how the CPU and memory work?
13. What are memory units of digital computers
14. ______ is a programmable machine.
15. Computers can execute a prerecorded list of instructions called______.
16. The actual machinery in a computer is called ______ and the instructions and the data are called_______.
17. Computer can be classified by their size and power as______________________.

18. ______ is a component of a computer that executes the instructions.
19. ______ allows a computer to permanently retain large amounts of data.
20. ______ enables a computer to store data and programs temporarily.
21. ______is conduit through which data and instructions enter a computer.
22. The number of bits that a computer can process at a time in parallel is called its_________.
23. CPU stands for
24. ALU stands for

1.3. Classification of Computers

Although a clear-cut classification of computers is difficult, computers can be broadly classified on the basis of purpose, components and size

1.3.1 Based on purpose:

i. Analog Computer: Analog Computers are computers that measure physical quantities like pressure, temperature, length etc., and convert them to numeric values. For example a thermometer does not perform any calculations but measures the temperature of the body by comparing the relative expansion of mercury.

Uses: Analog Computers are used mainly for scientific and engineering purposes because they deal with quantities that vary constantly. They give only approximate results.

ii. Digital Computers: Most computers are digital devices i.e. they count the numbers (or digits) that represent numerals, letters or other special symbols.

Uses: Special purpose digital computers can be fixed permanently in a machine. For example, processors that are installed in automobiles to control fuel, Braking system etc.

General-purpose digital computers can be used in different applications since they can store different sets of instructions and programs.

iii. Hybrid Computers: The features of analog and digital computers are combined to create a hybrid computing system. For example, analog devices measure a patient’s vital signs like temperature, heart functions. These are then converted to numbers and supplied to the digital components that monitor the patient's vital signs. Any fluctuation can thus be noticed immediately.

1.3.2 Based on power and size

i. **Microcomputer (Personal computers):** The microcomputer is the smallest type of computer available. Inside a microcomputer, the arithmetic and control units are combined on a single chip called a microprocessor. Microcomputers are used as personal computers by business executives where volumes of data processing and speed requirements are small.

ii. Minicomputer: Minicomputers are more powerful than the microcomputers and can support several users. They have larger RAM and backing storage capacity and can process data more quickly.

Minicomputers works well in what is known as Distributed Data Processing (DDP). That is, a company's processing power is decentralized, or distributed across different computers.

The first popular minicomputer was the PDP-8, launched in 1965. Minicomputers were basically developed for use in process control systems. Examples of popular minis are PDP-11 and the VAX 7500.

iii. Mainframe: The earliest computers were called mainframes due to their large size. The term is still used for the large computers of today. The capacities of the earlier mainframes and the mainframes of today are enormously different. Today's mainframes are far more powerful, so much so that a microcomputer today comes close to the processing power of an earlier mainframe.

Mainframes are very large computers with a very high capacity of main store. Because they can process large amounts of data very quickly, they are used by big companies, banks, and government departments as their main computer. They can be linked into a network with smaller departmental computers, microcomputers or with each other. They act as hosts of large national and international communication networks handling hundreds of users.

Example: IBM4381, ICL39 series and CDC Cyber series.

iv. Supercomputer: Complex scientific applications like weather forecasting require a large amount of data to be manipulated within a very short time. Large supercomputers with faster processing using multiple processors and superior technology are used for complex tasks requiring a lot of computational power. Super computers use "Non-Von-Neumann Designs."

Examples of Supercomputers are CRAY XMP-24 and NEC-500, PARAM PADMA

Portable Computers: Personal computing market is seeing the miniaturization phenomena. Now, computers are becoming smaller yet more powerful. One type of PC that is rapidly growing in popularity is the portable computer, which can be easily carried around. There are three categories of portable computers: Laptops or notebook PCs, Sub notebooks and Personal Digital Assistants.

1.3.3 Based on components

The earliest computers were mechanical computing devices. However, over generations the computer has developed from a huge mechanically operated device to convenient, portable desktops and laptops that function on the basis of small chips and integrated circuitry.

The first generation computers used vacuum tubes to store data and programming instructions. However vacuum tubes consumed a large amount of electricity, produced large amounts of heat, and was relatively unreliable.

The vacuum tubes were replaced by solid-state transistors. The second generation of computers was thus characterized by the use of transistors in place of vacuum tubes.

In the third generation of computers integrated circuits on chips (thin wafers of silicon) were used to store data and process instructions. These computers could handle more than one operation simultaneously.

The more circuits on a single chip, greater the amount of data that can be stored on the memory chip. While the technique of integrating circuits on a single chip was established during the third generation of computers, it was in the fourth generation of computers that Large Scale Integration (LSI) and Very Large Scale Integration (VLSI) of transistors on a single chip were used.

It is now possible to put the entire processor of the computer on a single chip. Intel Corporation developed the first Intel 4004 microprocessor chip. A microcomputer is built using a microprocessor chip, along with some other chips and circuitry.

The fifth generation of computers is conceived as a knowledgeable information processing system. It incorporates Artificial Intelligence.

SAQ

1. What are the different categories of digital computers?
2. Distinguish among the four kinds of computer system.
3. How is a PC different from a workstation?
4. What are the different types of portable computers?
5. What are the major functions of a computer?
6. What are mini computers?
7. What is the different between minicomputer and a microcomputer?
8. What is a mainframe and what is its typical application?
9. What are supercomputers and where they used?
10. The four categories of digital computer are_______, _______, _______ and _______.
11. _______is the smallestamong the portable computers.

1.4 Memory Device

Memory units are the internal storage areas in a computer. The term "memory" identifies data storage that comes in the form of chips, and the word "storage" is used for memory that exists on tapes or disks. Moreover, the term memory is usually used as a short form for physical memory, which refers to the actual chips capable of holding data. Some computers also use virtual memory, which expands physical memory onto a hard disk.

Every computer comes with a certain amount of physical memory, usually referred to as the main memory or the RAM. One can think of the main memory as an array of boxes, each of which can hold a single byte of information. A computer that has 1 megabyte of memory, therefore, can hold about 1 million bytes (or characters) of information. There are several different types of memory.

RAM Pronounced ram, acronym for Random Access Memory, a type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching that preceding bytes. RAM is volatile i.e its contents are lost when the device is powered off. RAM is the most common type of memory found in computers and other devices, such as printers.

There are two basic types of RAM:

- Dynamic RAM (DRAM)
- Static RAM (SRAM)

The two types differ in the technology they use to hold data, dynamic RAM being the more common type. Dynamic RAM needs to be refreshed thousands of times per second. Static RAM needs to be refreshed less often, which makes it faster; but it is also more expensive than dynamic RAM. Both types of RAM are volatile, meaning that they lose their contents when the power is turned off.

ROM: Pronounced rom, acronym for Read Only Memory and a computer memory on which data has been prerecorded. Once data has been written onto a ROM Chip, it cannot be removed and can only be read. Unlike the main memory (RAM), ROM retains its contents even when the computer is turned off. ROM is referred to as nonvolatile i.e. its contents are retained even when the device is powered off..

Most Personal computers contain a small amount of ROM that stores critical programs such as the program that boots the computer. In addition, ROMs are used extensively in calculators and peripheral devices such as laser printers, whose fonts are often stored in ROMs. A variation of a ROM is the PROM (programmable read-only memory). PROMs are manufactured as blank chips on which data can be written.

PROM: Pronounced prom, an acronym for programmable read-only memory. A PROM is a memory chip on which data can be written only once. Once a program has been written onto a PROM, it remains there forever. Unlike the main memory, PROMs, retain their contents when the computer is turned off.

The difference between a PROM and ROM (read-only memory) is that a PROM is manufactured as blank memory, whereas a ROM is programmed during the manufacturing process. To write data onto a PROM chip, one needs a special device called a PROM programmer or a PROM burner. The process of programming a PROM is sometimes called burning the PROM.

EPROM: Acronym for Erasable Programmable Read-Only Memory, and pronounced ee-prom, EPROM is a special type of memory that retains its contents until it is exposed to ultraviolet light. The ultraviolet light clears its contents, making it possible to reprogram the memory. An EPROM differs from a PROM in that a PROM can be written to only once and cannot be erased. EPROMs are used widely in personal computers because they enable the manufacturer to change the contents of the PROM before the computer is actually shipped. This means that bugs can be removed and new versions installed shortly before delivery.

EEPROM: Acronym for electrically erasable programmable read-only memory. Pronounced double-ee-prom, an EEPROM is a special type of PROM that can be erased by exposing it to an electrical charge. Like other types of PROM, EEPROM retains its contents even when the power is turned off. Also like all other types of ROM, EEPROM is not as fast as RAM. EEPROM is similar to flash memory (sometimes called flash EEPROM). The principal difference is the EEPROM requires data to be written or erased one byte at a time whereas flash memory allows data to be written or erased in blocks. This makes flash memory faster.

FLASH MEMEORY: Flash memory is a special type of EEPROM that can be erased and reprogrammed in blocks instead of one byte at a time. Many modern PCs have their BIOS (Basic Input Output System) stored on a flash memory chip so that it can easily be updated if necessary. Such a BIOS is sometimes called flash BIOS. Flash memory is also popular in moderns because it enables the modern manufacturer to support new protocols as they become standardized.

Auxiliary Storage Devices: Auxiliary storage also known as auxiliary memory or secondary storage, is the memory that supplements the main storage. This is a long-term, non-volatile memory. The term non-volatile means it stores and retains the programs and data even after the computer is switched off. Unlike RAM which looses the contents when the computer is turned off, and ROM, to which it is not possible to add anything new, auxiliary storage devices allows the computer to record information semi-permanently, so it can be read later by the same computer or by another computer. Auxiliary storage devices are also useful in transferring data or programs from one computer to another. They also function as back-up devices which allow to back-up the valuable information that you are working on. So even if by some accident your computer crashes and the data in it is unrecoverable, you can restore it from your back-ups. The most common types of auxiliary storage devices are magnetic tapes, magnetic disks, floppy disks, hard disks,

There are two types of auxiliary storage devices. This classification is based on the type of data access: sequential and random. Based on the type of access they are called sequential – access media and random-media. In the case of sequential-access media, the data stored in the media can only be read in sequence and to get to a particular point on the media you have to go through all the preceding points.

Magnetic tapes are examples of sequential – access media. In contrast, disks are random-access also called direct-access media because a disk drive can access any point at random without passing through intervening points. Other examples of direct-access media are magnetic disks, optical disks, zip disks, etc.

Magnetic Tape: Magnetic tapes is a magnetically coated strip of plastic on which data can be encoded. Tapes for computers are similar to the tapes used to store music. Storing data on tapes is considerably cheaper than storing data on disks. Tapes also have large storage capacities, ranging from a few hundred kilobytes to several gigabytes.

Accessing data on tapes, however, is much slower than accessing data on disks. Tapes are sequential –access media, which means that to get to a particular point on the tape, the tape must go through all the preceding points. In contrast, disks are random – access media because a disk drive can access any point at random without passing through intervening points.

Because tapes are so slow, they are generally used only for long-term storage and backup. Data to be used regularly is almost always kept on a disk. Tapes are also used for transporting large amounts of data. Tapes come in a variety of sizes and formats. Tapes are sometimes called streamers or streaming tapes.

Hard Disk: Hard disk is a magnetic disk on which user can store computer data. The term hard is used to distinguish it from a soft, or floppy, disk. Hard disks hold more data and are faster than floppy disks. A hard disk, for example, can store anywhere from 10 megabytes to several gigabytes, whereas most floppies have a maximum storage capacity of 1.4 megabytes.

A Single hard disk usually consists of several platters. Each platter requires two read/write heads, one for each side. All the read/write heads are attached to a single access arm so that they cannot move independently. Each platter has the same number of tracks, and a track location that cuts across all platters is called a cylinder. For example, a typical 84-megabyte hard disk for a PC might have two platter (four sides) and 1,053 cylinders.

Floppy Disk: Floppy disk is a soft magnetic disk. Floppy disks are portable. Disk drives for floppy disks are called floppy drives. Floppy disks are slower to access than hard disks and have less storage capacity, but they are less expensive.

Floppies come in two basic sizes:

- **5 ¼ -inch –** The common size for PCs made before 1987. This type of floppy is generally capable of storing between 100K and 1.2MB (megabytes) of data. The most common sizes are 360K and 1.2MB.
- **3 ½ -inch** –They are encased in a rigid envelope. The most common sizes for PCs are 720K (double-density) and 1.44MB (high-density).

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 –

up to 6 gigabytes (6 billion bytes) – than magnetic media, such as floppies and hard disks. There are three basic types of optical disks:

- **CD – ROM -** Like audio CDs, CD-ROMs come with data already encoded onto them. The data is permanent and can be read any number of times, but CD-ROMs cannot be modified.
- **WORM** – This term stands for write-once, read-many. With a WORM disk drive, you can write data onto a WORM disk, but only once. After that, the WORM disk behaves just like a CD-ROM.
- **Erasable** – Optical disks that can be erased and loaded with new data, just like magnetic disks. These are often referred to as EO (erasable optical)disks.

These three technologies are not compatible with one another; each requires a different type of disk drive and disk. Even within one category, there are many competing formats, although CD-ROMs are relatively standardized.

CD-ROM: CD-ROM, which is pronounced as 'see-dee-rom', is the abbreviation of compact Disc-Read-only Memory. CD-ROM is a type of optical disk capable of storing large amounts of **data** – up to 1 GB, although the most common size is 630MB (megabytes). A single CD-ROM has the storage capacity of 700 floppy disks, enough memory to store about 300,000 text pages.

CD-ROMs are recorded by the vendor, and once recorded, they cannot be erased and filled with new data. To read a CD, one needs a CD-ROM player. Also called a CD-ROM drive, a CD-ROM player is a device that can read information from a CD-ROM. CD-ROM players can be either internal, in which case they fit in a bay, or external, in which case they generally connect to the computer's parallel port.

Parallel CD-ROM players are easier to install, but they have several disadvantages: They're somewhat more expensive than internal players they use up the parallel port which means that the port can't be used for another device such a printer, and the parallel port itself may not be fast enough to handle all the data pouring through it.

SAQ

1. What are the different types of memory?
2. What is the difference between RAM and ROM?
3. What are the two basic types of RAM?
4. What is PROM?
5. What is EPROM?
6. How is PROM different from EPROM?
7. What is EEPROM?
8. How EEPROM is different from EPROM and PROM?
9. What is flash memory?
10. What is an auxiliary storage device?
11. What are the major auxiliary storage devices for a computer?
12. What is the difference between a random and a sequential access devices?
13. What is a magnetic tape and how does it work?
14. What are the advantages and disadvantages of magnetic tapes?
15. What are the different types of magnetic tapes?
16. What is a hard disk and how does it work?
17. What are optical disks and how to they work?
18. What is a floppy disk how is it different from a hard disk and what are the different types?
19. RAM stands for ___________________.
20. The memory unit that holds instructions for starting up the computer is ____________.
21. _______________Is a memory chip on which programs can be stored
22. There are two types of auxiliary storage devices: ______________ and _____________.
23. Magnetic tape is an example of ___________________media.
24. ________a type of magnetic tape that uses the same technology as VCR tapes.
25. DAT stands for ___________________.
26. ________________is a high – capacity floppy disk drive developed by Iomega Corporation.
27. The removable cartridges of a Jaz drive can hold _________________ of data.

28. _______________are a storage medium from which data is read and to which it is written by lasers.

1.5 Input Devices

An input device is any machine that feeds data into a computer. For example, a keyboard is an input device, whereas a display monitor is an output device. Input devices other than the keyboard are sometimes called alternate input devices. Mice, trackballs, and light pens are all alternate input devices.

a) Keyboard: Keyboard is an input device consisting of a set of typewriter-like keys that enable users to enter data into a computer. Computer keyboards are similar to electric-typewriter keyboards but contain additional keys. The keys on computer keyboards are often classified as follows:

- Alphanumeric keys – letters and numbers
- Punctuation keys – comma, period, semicolon, and so on.
- Special keys – function keys, control keys, arrow keys, Caps Lock key

The standard layout of letters, numbers, and punctuation is known as a QWERTY keyboard because the first five keys on the top row of letters spell QWERTY. The QWERTY Keyboard was designed in the 1800s for mechanical typewriters and was actually designed to slow typists down to avoid jamming the keys. Another keyboard design, which has letters positioned for speed typing, is the Dvorak keyboard.

There is no standard computer keyboard, although many manufacturers imitate the keyboards of PCs. There are actually three different PC keyboards: the Original PC keyboard, with 84 keys; the AT Keyboard, also with 84 keys; and the enhanced Keyboard, with 101 keys. The three differ somewhat is the placement of function keys, the Control key, the Return key, and the Shift keys.

b) Mouse: Mouse is a device that controls the movement of the cursor or pointer on a display screen. A mouse is a small object that can be rolled along a hard, flat surface. As the name indicates, it looks a bit like a mouse. As the mouse is moved, the pointer on the display screen moves in the same direction. Mice contain at least one button and sometimes as many as three, which have different functions depending on what program is running.

The mouse is one of the great breakthroughs in computer ergonomics because it frees the user to a large extent from using the keyboard. In particular, the mouse is important for graphical user interfaces because user can simply point to options and objects and click a mouse button. Such applications are often called point – and click programs. .

Types of Mice
There are three basic types of mice.

i. Mechanical: It has a rubber or metal ball on its underside that can roll in all directions. Mechanical sensors within the mouse detect the direction the ball is rolling and move the screen pointer accordingly.

ii. Opto-mechanical: Same as a mechanical mouse, but uses optical sensors to detect motion of the ball.

iii. Optical: It uses a laser to detect the mouse's movement. This mouse must be moved along a special mat with a grid so that the optical mechanism has a frame of reference. Optical mice have no mechanical moving parts. They respond more quickly and precisely than mechanical and opto mechanical mice, but they are also more expensive.

c) Trackball

Trackball is another pointing device. Essentially, a trackball is a mouse lying on its back. To move the pointer the ball is rotated with thumb. There are usually one to three buttons next to the ball, which is used just like mouse buttons.

The advantage of trackballs over mice is that the trackball is stationary so it does not require much space to use it. In addition, trackball can be placed on any type of surface, including lap. For both these reasons, trackballs are popular pointing devices for portable computers.

d) Scanners

Scanner is an input device that can read text or illustrations printed on paper and translate the information into a form that the computer can use. A scanner works by digitizing an image – dividing it into a grid of boxes and representing each box with either a zero or a one, depending on whether the box is filled in. (For color and gray scaling, the same principle applies, but each box is then represented by up to 24 bits.) The resulting matrix of bits, called a bit map, can then be stored in a file, displayed on a screen, and manipulated by programs.

Optical scanners do not distinguish text from illustrations; they represent all images as bit maps. Therefore, one cannot directly edit text that has been scanned. To edit text read by an optical scanner, one needs an optical character recognition (OCR) system to translate the image into ASCII characters. Most optical scanners sold today come with OCR packages.

e) Magnetic Ink Character Recognition (MICR)

Magnetic Ink Character Recognition (MICR) allows the computer to recognize characters printed using magnetic ink. MICR is a direct-entry method used in banks. This technology is used to automatically read those frustrating-looking numbers on the bottom of the cheque. A special –purpose machine known as a reader/sorter reads characters made of ink containing magnetized particles. A related technology is the magnetic strip, used on the back of credit cards and bank debit cards, that allows readers such as Automated Teller Machines (ATMs) to read account information and facilitate monetary transactions. Another example of magnetic strip technology is in ID cards, which can be used for a variety of functions from attendance monitoring to restricting access to specific locations.

e) Optical Mark Recognition(OMR) Optical Mark Recognition (OMR) also called mark sensing is a technology where an OMR device senses the presence or absence of a mark, such as a pencil mark. OMR is used in tests such as aptitude tests.

f) Bar Code Reader

Bar-code readers are photoelectric scanners that read the bar codes, or vertical zebra striped marks, printed on product containers. Supermarkets user a bar code system called the Universal Product Code (UPC).

SAQ

1. What are the different kinds of input devices?
2. What are the different point devices available?
3. What is the difference between a mouse and a trackball?
4. What is a Joystick and what is its most common user?
5. What is the function of a scanner and what are the different types of scanners?
6. Explain MICR, OCR and OMR?
7. . ______________is the input device consisting of a set of keys that enables you to enter data into a computer.
8. A device that controls the movement of the cursor or pointer on a display screen is called the ________________.
9. Mouse was invented By _______________ in 1963.
10. The three basic types of mice are ________________, _____________ and __________
11. ______________ are more popular than the mouse with portable computers.
12. The input device used mostly for computer games is the _____________.
13. The input device that enables you to enter drawings and sketches into a computer is ________________.
14. _____________ is a n input device that can read text or illustrations printed on paper and translate the information into a form that the computer can use.
15. __________ Allows the computer to recognize characters printed using magnetic ink.

1.6 output Devices

An output device is any machine capable of representing information from a computer. Output devices include display screens, loudspeakers, printers, plotters, etc.

a) Monitor : Monitor is another term for the display screen. The term monitor, however, usually refers to the entire box, whereas display screen can mean just the screen. There are many ways to classify monitors. The most basic is in terms of colour capabilities, which separates monitors into three classes:

Monitors are classified as 1.monochrome 2. Colour based on colour capabilities.

1.Analog and Digital based on signals

i. Monochrome : Monochrome monitors actually display two foregrounds. The colours can be black and white, green and black, or amber and black.
ii. Grey Scale:A gray – scale monitor is a special type of monochrome monitor capable of displaying different shades of gray.

ii. Colour : Colour monitors can display anywhere from 16 to over 1 million different colours. Colour monitors are sometimes called RGB monitors because they accept three separate signals – red, green, and blue. All colour computer monitors are RGB monitors. An RGB monitor consists of a vacuum tube with three electron guns – one each for red, green, and blue at one end and the screen at the other end. The three electron guns fire electrons at the screen, which contains a phosphorous coating. When the electron beams excite the phosphors, they glow. Depending on which beam excites them, they glow red, green, or blue. Ideally, the three beams should converge for each point on the screen so that each pixel is a combination of the three colours.

b) Printers: Printer is a device that prints text or illustrations on paper and in many cases on transparencies and other media. There are many different types of printers.
Printers are also classified according to the following characteristics:

- **Quality of type -** The output produced by printers is said to be either letter quality (as good as a typewriter), near letter quality, or draft quality. Only daisy – wheel, ink – jet, and laser printers produce letter – quality type. Some dot – matrix printers claim letter – quality print, but if you look closely, you can see the difference.

- **Speed –** Measured in characters per second (cps) or pages per minute (ppm), the speed of printers varies widely. Daisy – wheel printers tend to be the slowest, printing about 30 cps. Line printers are faster (up to 3,000 lines per minute). Dot – matrix printers can print up to 500 cps, and laser printers range from about 4 to 20 text pages per minute.

- **Graphics –** Some printers (daisy – wheel and line printers) can print only text. Other printers can print both text and graphics.

- **Fonts –** Some printers, notably dot – matrix printers are limited to one or a few fonts. In contrast, laser and ink – jet printers are capable of printing an almost unlimited variety of fonts. Daisy – wheel printers can also print different fonts, but you need to change the daisy wheel, making it difficult to mix fonts in the same document.

In terms of the technology utilized, printers fall into two categories impact and non impact.

Impact printers: An impact printer needs to physically strike a ribbon against the paper to make a mark, which is where the name comes from. In the early days of the computer era, the best print quality came form so-called daisy-wheel printers, which worked pretty much like a typewriter operated by a superhumanly fast typist. They printed beautifully, but they were slow, couldn't do graphics and all were noisy. Impact printers are daisy-wheel printer and Dot-matrix printer.

i. **Daisy – wheel Printer:** Daisy – wheel printers are a type of printer that produces letter – quality type. A daisy – wheel printer works on the same principle as a ball – head typewriter. The daisy wheel is a disk made of plastic or metal on which characters stand out in relief along the outer edge.

To print a character, the printer rotates the disk until the desired letter is facing the paper. Then a hammer strikes the disk, forcing the character to hit an ink ribbon, leaving an impression of the character on the paper. One can change the daisy wheel to print different fonts. Daisy – wheel printers cannot print graphics, and in general they are noisy and slow, printing from 10 to about 75 characters per second. As the price of laser and ink-jet printers has declined, and the quality of dot – matrix printers has improved, daisy – wheel printers have become almost obsolete.

ii. Dot – matrix Printer: Dot – matrix printers create characters by striking pins against an ink ribbon. Each pin makes a dot, and combinations of dots form characters and illustrations. Dot – matrix printers are inexpensive and relatively fast, but they do not produce high – quality output. Dot – matrix printers vary in two important characteristics:

- **Speed -** Given in characters per second (cps), the speed can vary from about 50 to over 500 cps. Most dot – matrix printers offer different speeds depending on the quality of print desired.
- **Print quality –** Determined by the number of pins (the mechanisms that print the dots0, it can vary from 9 to 24. The best dot – matrix printers (24 pints) can produce near letter – quality type, although one can still see a difference if you look closely.

In addition to these characteristics, one should also consider the noise factor. Compared to laser and ink – jet printers, dot – matrix printers are notorious for making a racket. In addition, since they are impact printers, the dot – matrix printers can print to multi – page forms (that is, carbon copies), something laser and ink – jet printers cannot do.

Non-impact printer: A non-impact printer is a type of printer where the print head does not strike the paper. These types of printers used almost everywhere now, are faster and quieter than impact printers because they have fewer moving parts. Non impact printers form characters and images without direct physical contact between the printing mechanism and the paper. The three main types of non-impact printers include inkjet, laser and thermal printers.

i. Ink jet Printer: Ink – jet printers work by spraying ionized ink at a sheet of paper. Magnetized plates in the ink's path direct the ink onto the paper in the desired shapes. Ink – jet printers are capable of producing high quality print approaching that produced by laser printers. A typical ink – jet printer provides a resolution of 300 dots per inch, although some newer models offer higher resolutions.

In general, the price of ink – jet printers is lower than that of laser printers. However, they are also considerably slower. Another drawback of ink – jet printers is that they require a special type of ink that is apt to smudge on inexpensive copier paper.

Because ink – jet printers require smaller mechanical parts than laser printers they are especially popular as portable printers. In addition, colour ink – jet printers provide an inexpensive way to print full – colour documents.

ii. Laser Printer: Laser printer utilizes a laser beam to produce an image on a drum. The light of the laser alters the electrical charge on the drum wherever it hits. The drum is then rolled through a reservoir of toner, which is picked up by the charged portions of the drum. Finally, the toner is transferred to the paper through a combination of heat and pressure. This is also the way copy machines work.

Because an entire page is transmitted to a drum before the toner is applied, laser printers are sometimes called page printers.

One of the chief characteristics of laser printers is their resolution – how many dots per inch (dpi) they lay down. The available resolutions range from 300 dpi at the low end to 1,200 dpi at the high end.

In addition to the standard monochrome laser printer, which uses a single toner, there also exist colour laser printers that use four toners to print in full colour. Colour laser printers tend to be about five to ten times as expensive as their monochrome siblings.

Laser printers produce very high – quality print and are capable of printing an almost unlimited variety of fonts. Because laser printers are non – impact printers, they are much quieter than dot – matrix or daisy – wheel printers. They are also relatively fast, although not as fast as some dot – matrix printers. The speed of laser printers ranges from about 4 to 20 pages of text per minute (ppm). A typical rate of 6 ppm is equivalent to about 40 characters per second (cps).

iii. Thermal Printer: Thermal printers are printers that produce images by pushing electrically heated pins against special heat – sensitive paper. Thermal printers are inexpensive and are used in most calculators and may fax machines. They produce low – quality print, and the paper tends to curl and fade after a few weeks or months.

Plotters: Plotter is a device that draws pictures on paper based on commands from a computer. Plotters differ from printers in that they draw lines using a pen. As a result, they can produce continuous line, whereas printers can only simulate lines by printing a closely spaced series of dots. Multi colour plotters use different – coloured pens to draw different colours. In general, plotters are considerably more expensive than printers. They are used in engineering applications where precision is mandatory.

1.7 Summary

1. A computer is an electronic device with the ability to accept data and instructions. Process the data, store them and produces output as the result.
2. The main peripherals of a computer are input system, CPU, output system.
3. CPU - Central Processing Unit, which consists of Arithmetic and Logic Unit(ALU), Central Unit(CU) and Memory Unit (MU).
4. Based on purpose computers are classified into Analog, Digital and Hybrid computers
5. Based on power and size computers are classified into Micro computer, Mini Computer, Main Frame and Super computers.
6. Memory units are the internal storage areas in a computer.
7. RAM: Random Access Memory which is volatile in nature.
8. ROM: Read Only Memory, this is non-volatile in nature.
9. An input device is any machine that feeds data into a computer for example keyboard, mouse, joy sticks and so on.
10. An output device is any machine capable of representing information from a computer for example monitors printers etc.

SAQ

1. Define computer?
2. List characteristic of computer?
3. Explain classification of computers?
4. List the difference of Input and Output devices?
5. Briefly explain about memory unit?

UNIT - II
OPERATING SYSTEMS

Structure

2.0 Objectives

At the end of this unit student will be able to define operating system, its functions, types of operating systems, internal and external commands.

2.1 Introduction

Software is the part of the computer system which enables the hardware to operate. Computer Software can be divided into two major classifications

a. System Software
b. Application Software

System Software includes the computer programs that run a computer system itself or that assist a computer in running application programs. It also includes the documentation that describes how these programs operate.

Application Software included the computer programs and related documentation that accomplish end-user data processing tasks.

System Software consists: a) Operating system b) Utilities.

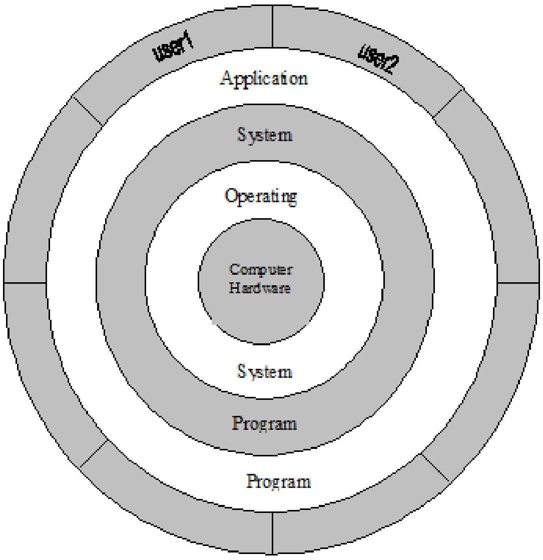
The student has to thoroughly study and understand concepts explained in this unit for the following reasons.

1. User interacts with the computer through operating system in order to accomplish his/her task since it is his primary interface with a computer.
2. Many concepts and techniques found in operating system have general applicability in other applications.

Basics of Operating System

An operating system is an essential component of a computer system. The primary objective of an operating system is to make computer system convenient to use and utilize computer hardware in an efficient manner. In other words operating system is defined as the interface between user and the peripherals of the computer.

An operating system is a large collection of software, which manages resources of the computer system, such as memory, processor, and file system and input/output devices. It keeps track of the status of each resource and decides who will have a control over computer resources, for how long and when. The positioning of operating system in overall computer system is shown in figure.



Component of computer system

The Operating system software must keep track of each hardware resource, determine who gets what, determine when the user will have access to the resource, allocate how much of the resource the user will be given, and terminate access at the end of the use period.

Operating systems vary in complexity from those that support single user microcomputers to those that handle multi-user mainframes. Their complexity depends on the computer system's size and scope, together with the type of performance provided to its users. A single stand-alone microcomputer will have a relatively simple operating system, whereas a mainframe that supports hundred of users accessing the system simultaneously will have one that is far more complex.

The primary purpose of an operating system is to maximize the productivity of a computer system by operating it in the most efficient manner and minimizing the amount of human intervention required. An operating system also simplifies the job of computer programmers, since it includes programs that perform common input/output and storage operations and other standard processing functions

Many operating systems are designed as a collection of program modules, which can be organized in combination with various capabilities around a central module, or kernel. Such operating systems can be tailored to fit the processing power and memory capability of a computer system and the type of processing jobs that need to be done on it.

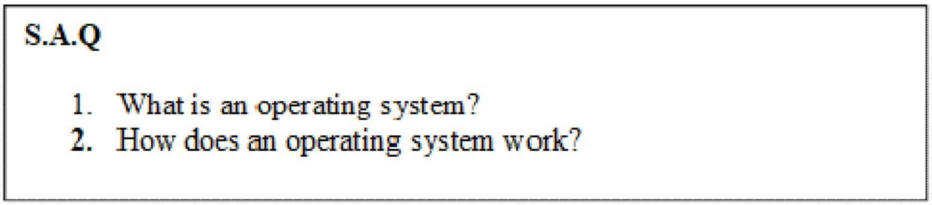
- An operating system is an essential component of a computer system.
- The primary objective of an operating system is to make computer system convenient to use and utilize computer hardware in an efficient manner.
- An operating system (OS) is an integrated set of specialized programs which permit the continuous operation of a computer from one program to the next with the minimum amount of operator intervention.

Through the Operating system the computer can supervise its own operating by automatically calling in the applications programs, translating any other special service programs, and managing the data to produce the desired output. There are two ways one can interact with operating systems:

1. Directly by means of Operating System Commands
2. By means of operating System Call in a program.

Operating System Commands: Apart from system calls, users may interact with operating system directly by means of operating system commands. When user invokes Operating System Command, the operating system acts as an interface between user and the hardware of a computer system by bringing the controlling and allocating resources together according users requirement.

System Call: System calls provide the interface to a running program and the operating system. User Program receives operating system services through the set of system calls.



S.A.Q

1. What is an operating system?
2. How does an operating system work?

2.3 Functions of Operating System

Even the simplest operating system in a minicomputer or mainframe performs a number of resource management tasks or functions. These functions include job management, batch processing, on-line processing, data management, virtual storage, and input/output management.

a. **Job Management:** Job management software manages the jobs waiting to be processed. It recognizes the jobs, identifies their priorities, determines whether the appropriate main memory and secondary storage capability they require is available, and schedules and finally runs each job at the appropriate moment.

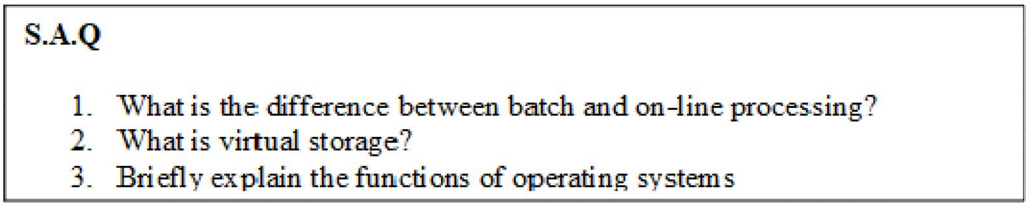
b. **Batch Processing:** System software is available to support the different methods of processing a job with batch processing, the most basic method, and data are accumulated and processed in groups. Payroll applications, for example, are often processed this way. Once in every week, hourly records are grouped and the payroll software is run.

c. **On-line Processing:** In on-line processing, data are processed instantaneously. For example, a sales person may need to find out whether a particular item requested by a customer is in stock for immediate shipment. Using an on-line system the request for information will be taken to access the central database and return the requested information to the terminal from which the request was made. All of these steps take less than a few seconds, at the most. Most on-line operating systems have multi-user and multitasking capabilities.

d. **Data Management:** In the process of managing the resources of the computer system, operating system software also manages the storage and retrieval of data. As the system software handles many of the details associated with this process, such details are not a primary concern for users or programmers writing application programs.

e. **Virtual Storage:** Operating systems also manages the allocation of main memory to specific jobs. Some operating systems have a feature called virtual storage. With this software it is possible to increase the capacity of main memory without actually increasing its size. This is accomplished by breaking a job into sequences of instructions, called pages or segments, and keeping only a few of these in main memory at a time; the Reminder are kept on secondary storage devices. As a result, relatively large jobs can be processed by a CPU that in fact contains a relatively small memory.

f. **Input/Output Management:** Operating systems also manage the input to and output from a computer system. This applies to the flow of data among computers, terminals, and other devices such as printers. Application programs use the operating system extensively to handle input and output devices as needed. For large systems, the operating system has even greater responsibilities and powers. It is like a traffic policeman – it makes sure that different programs and users running at the same time do not interfere with each other. The operating system is also responsible for security, ensuring that unauthorized users do not access the system.



S.A.Q

1. What is the difference between batch and on-line processing?
2. What is virtual storage?
3. Briefly explain the functions of operating systems

2.4 Classification of Operating Systems

The classification of an operating system is a grouping that differentiates or identifies the operating system based on how it works the type of hardware it controls and the applications it supports. Operating systems can be classified as follows:

a. **Time Sharing Operating System:** A time sharing system allows many users to share the computer resources simultaneously. In other words, time sharing refers to the allocation of computer resources in time slots to several programs simultaneously. In an interactive system, many users directly interact with the computer form terminals connected to the computer system. They submit small execution requests to the computer and expect results back immediately, after a short enough delay to satisfy their temperament. We need a computer system that supports both multi programs and multi processes. The process appears to be executing simultaneously, each at its own speed. Apparent simultaneous execution of processes is achieved by frequently switching the processor from one process to another in a short span of time. These systems are often called time–sharing systems.

b. **Multi-user:** Multi-user operating systems allow two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of concurrent users. The operating systems of mainframes and minicomputer are multi-user systems. Examples are MVS, UNIX, etc. Another term for multi-user is time-sharing.

c. **Multiprocessing:** Multiprocessing refers to a computer system's ability to support more than one process (program) at the same time. Multiprocessing operating systems enable several programs to run concurrently. MVS and UNIX are two of the most widely used multiprocessing systems, but there are many others, including OS/2 for high-end PCs. Multiprocessing systems are much more complicated than single-process systems, because the operating system must allocate resources to competing processes in a reasonable manner. Multiprocessing also refers to the utilization of multiple CPUs in a single computer system. This is also called parallel processing

d. **Multitasking:** Multitasking allows more than one program to run concurrently. Multitasking is the ability to execute more than one task at the same time, a task being a program. The terms multitasking and multiprocessing are often used interchangeably, although multiprocessing sometimes implies that more than one CPU is involved. In multitasking, only one CPU is involved, but it switches from one program to another so quickly that it gives the appearance of executing all the programs at the same time. There are two basic types of multitasking: 1.Preemptive and 2.Cooperative In preemptive multitasking, the operating system parcels out CPU time slices to each program. In cooperative multitasking, each program can control the CPU for as long as it needs it. If a program is not using the CPU, however, it can allow another program to use it temporarily. OS/2, Windows 95, Windows NT, the Amiga operating system and UNIKX use preemptive multitasking, whereas Microsoft Windows 3.x and the MultiFinder (for Macintosh computers) use cooperative multitasking.

e. **Multithreading:** Multithreading allows different parts of a single program to run concurrently. Multithreading is the ability of an operating system to execute different parts of a program, called threads, simultaneously. The programmer must carefully design the program in such a way that all the threads can run at the same time without interfering with each other.

f. **Real-time:** Real-time operating systems are systems that respond to input immediately. This category includes operating systems designed substantially for the purpose of controlling and monitoring external activities with timing constraints. They are used for such tasks as navigation, in which the computer must react to a steady flow of new information without interruption. Most general-purpose operating systems like DOS and UNIX are not real-time because they can take a few seconds, or even minutes, to react. Some examples of real-time operating systems are:

Basic Real-time Monitor (Real-time OS for the Phillips P-855 and P-860)

BLMX (Board-level multitasking executive for National Semiconductor 8080 and Z80 based CPU boards)

BSO/RTOS (Small real-time OS for Z80, 6809, 8086, 68000. Boston Systems Office)

C executive (Memory-based real-time UNIX-like executive for the PDP-11 or VAX)

CCP (Communications Control Program. Limited OS for the IBM System/3)

CTOS (Real-time, multitasking, multiprocessing OS for 8086 family)

CTRON (Specification for a version of TRON for communication and network control)

2.5 Disk Operating System

Disk Operating System (DOS) that can use a disk storage device such as a floppy disk, hard disk drive or optical disc. A disk operating system must prove a file system for organizing, reading and writing files on the storage disk. When a computer is switched on POST (Power OnSelf Test).

- In the 1980s or early 1990s, the operating system that shipped with most PCs was a version of the Disk Operating System (DOS*)* created by Microsoft: MS-DOS.
- MS-DOS is a disk operating system for IBM PC–compatible computers.
- In its day, it was easily the most popular operating system in the world.
- As with any other operating system, its function is to oversee the operation of the system by providing support for executing programs, controlling I/O devices, handling errors, and providing the user interface.
- MS-DOS is a disk-based, single-user, single-task operating system. These qualities make it one of the easiest disk operating systems to understand.

DOS consists of an input/output system, a command processor and several utilities. The needed often enough to make it necessary or practical to keep them in the computer's RAM all the time. FORMAT.COM, the program that formats blank disks, is an example of a DOS utility. Sometimes these utilities are called external commands (as opposed to internal commands which are included as part of the file COMMAND.COM and remain resident in memory at all times; e.g., DIR and COPY).

The command processor is also a file you see on the disk, but once read into the computer's memory, it usually resides there. Some programs provide their own command processor, and there are times when the command processor will be overwritten in memory by a program and have to be reloaded when the program stops executing.

The input/output system consists of two files and a ROM (Read Only Memory) chip. While the two files are on your disks and are loaded into memory when the computer starts, they are normally hidden from your view and not available to you for changing.

Input/Output System: This most primitive of the DOS systems has two parts:

• BIOS (Basic Input/Output System). These are the fundamental routines that control the keyboard, video display and other peripherals. The BIOS is comprised of a ROM on the computer's main circuit board and the file IBMBIO.COM (or IO.SYS), one of the two hidden files on your disk.

Operating System : This is the main file-handling system for the computer. It consists of two systems one for disk-based files and the other for non-disk peripheral devices. They are in hidden file IBMDOS.COM (or MSDOS.SYS).

CommandProcessor

The command processor (COMMAND.COM) which performs three tasks:
• **It handles critical interrupts:** COMMAND.COM takes care of all demands for attention by parts of the computer. The user typing the Control-Break program break command is an example of an interrupt.

• **It handles critical errors:** COMMAND.COM takes care of problems. For example, if you leave the disk drive door open during a disk operation COMMAND.COM is responsible for the error message you will see.

• **It performs end-of-program housekeeping:** COMMAND.COM takes care of making the computer's memory available for other programs and reloading parts of itself ifthe program wrote over them.

COMMAND.COM also places the **C>** prompt on the screen and interprets any command(s) you might type.

DefaultDrive:

The default drive is the first disk drive on which DOS will look for a program if no drive specification is given with the filename. How do you know what it is? Look at the prompt. The default drive letter is part of the prompt (unless someone has changed the prompt to eliminate it).

A:\> indicates that drive A (the left or top drive in a two-drive system) is the default drive. The right (or second) drive in such a system is called drive **B** and the first hard disk in any system is given the letter **C** as its drive designation.

DOS supports many more than drives **A** through **C**. In fact, a computer can specify up to 63 drive names. To change to drive **C** type **C**: as shown here:

```
A: \ > C:

C: \ >
```

Rules for Filenames :

Like devices, disk files have to be identified so DOS can address them. These filenames have specific rules. The basic form of a filename is:

Filename.ext

The first part of the name to the left of the period is called the root name. The root name canbe from one to eight characters long and cannot be the same as a device name. The second part tothe right of the period is the extension. It is optional and, if used, can be one to three characters long. The period is used between the root name and extension and must be present if there is an extension.

The following are legal and illegal characters in a filename:

- **Legal:** A-Z 0-9 $#&@!()-{}'`_~

- **Illegal :**|<>\^+=?/[]";,* plus control characters and the space

Some other operating systems allow longer file names and there are commercial utilities which link a database of long names to your short names so you can find files by using more fully descriptive names.

Note: Windows allows longer file names with the space but underneath the facade the format is maintained.

Each DOS command has a mandatory part and some have an optional part. The mandatory parts will be shown in bold CAPITAL LETTERS and the optional parts in lower case.
DOS commands are issued at the prompt **C:\>** Whatever you type after that prompt that is not in the **COMMAND.COM** standard library is assumed to be the name of a file on the default disk and DOS will search for it under one of three names (in the order listed).

If you type **C\:> FILENAME**

DOS will look for: **FILENAME.COM** or **FILENAME.EXE** or **FILENAME.BAT**
The first is a command file (note the COM extension). The second is an execution file (EXEextension). And, the third is a batch file (a series of DOS commands in a text file which you'll learn about later in this tutorial).
The first file found will be read into memory and the command processor will start the program running.
Both .COM and .EXE files execute as programs. The difference between the two relates to how memory is allocated and certain parameters in the computer are set.

2.5.1 Internal commands

DOS commands for which the specifications are available in shell(command.com) are called Internal commands. These are frequently used commands and are called **resident commands.**

1. CLS: Clears the screen and puts the cursor in the home (upper left) position.

2. DIR: this command allows the user to see all files and sub-directory in the current directory. **DIR** command lists file information in five columns; first column gives primary name of the file, second column gives extension of the file, third column gives the file size – number of bytes used; fourth column gives the last updated date, and fifth column gives the last updated time.

DIR alone will show the complete directory. With the optional filename, DIR will try to find just that file.

- **Dir /p** option causes a pause when the screen fills.
- **Dir /w** option yields a full 80-column display of just the filenames.
- **Dir/s** displays all sub directory and files in the sub-directory of current directory.

There are other options for sorting the listing and displaying the contents of lower-level directories.

3.C:\>COPY CON: TEXTFILE

This is the text to go into the text file being created. Each line is typed to the screen and it is being saved into a buffer for later transfer to the file TEXTFILE. Each line may be corrected as it is typed, but cannot be changed after it is terminated by the carriage return. Also, if you happen to type beyond column 80 on the screen, you cannot correct anything on the line above. Each linemust be terminated by a carriage return (the enter key). You signal you are finished by typing a Control-Z, the symbol for end-of-file, followed by Return. ^Z .**1 File(s) copied**

4. COPY FILES

The **COPY** command is a very powerful command within DOS. With it you can create duplicates of individual files, join several files into one, and even use your computer like as simple typewriter by "copying" from the device named CON: to the device named PRN

Copying one file to another (copies from filename1 to filename2):

COPY d1:FILENAME1.ext d2:filename2.ext/v
/v option verifies the copy as it takes place. This adds confidence at the price of slower operation.
There are other options not shown here. Wildcards are allowed.

For example :
C:\>COPY ADDRS.LST B: Copies the single file **ADDRS.LST** from **C: to B:**.
C:\>COPY *.* B:/V Copies all files on **C:** to the disk in **B:** and verifies the information is it is being written.

C:\>COPY ADDRS.LST Yields an error message. Can't copy a file to itself.

C:\>COPY B:*.* Copies all files from drive B: to drive C:. (If a destination is not specified, the default drive and directory is used.) Copy can also be used to concatenate (join) several files by using the following form:

COPY d1:FILENAME1.ext+d2:FILENAME2.ext+... d0:filename0.ext/v

The options are the same as the previous version of the copy command.
All specified filenames (#1, #2, etc.) will be copied and joined into filename0.
If no filename0 is specified, the first source file named will be used.
Wildcards are dangerous with this command.

Example:
Contents of **FILE1:** This is file number one
Contents of **FILE2:** This is file number two

C:\>COPY FILE1+FILE2 FILE3

Contents of **FILE3:** This is file number oneThis is file number two The **COPY** command can be used to create text files by copying from device **CON:** to a file.

5. DATE AND TIME :

These two commands show and/or set the system date and time. New computers have the clock built-in and do not require a program to load the time. If your clock battery fails, the default values will be 13-8-18 for the date and 00:00:00.00 for time. Now and again you will see files with a create date of 13/8/18; they were created on a system where the clock has failed and DOS has used its default value. For the DATE command you can enter the date as month/day/year with hyphens or slashes, i.e., 13/8/18 or 13-8-18 are acceptable dates. Do not enter the day of the week, even though it shows on the screen. The computer will calculate it for you.

The format for DATE is:

DATE <date> : On early computers the time setting required a 24-hour clock, i.e., any time after noon had to have 12 added to it, for example 3:00 pm had to be entered as 15:00. While the TIME command will still respond to this type of time, you may not also enter 3:00p and the computer is smart enough to know you mean 15:00.

The format for TIME is:

TIME <time> : On most computers these commands will change the permanent clock settings as well as changing the date/time in DOS.

6. MD (MKDIR)

It creates a new directory in sub directory in the current directory.

Syntax: **C:\> MD<new directory name>**

7. RD(RMDIR)

This command is used to remove a directory from the dist. It can’t remove a directory which contains subdirectory or files, ie the child should be removed from the parent. Similarly this command can’t remove the current directory and root directory.

Syntax: **C:\>RD<DIR NAME>**

8. TYPE

It displays the content of saved file. Syntax: **C:\> TYPE > FILE NAME**

9. REN

This command changes the name of existing file or directory

Syntax: **C:\> REN<OLD NAME><NEW NAME>**

10. DELETE

Delete a file form current directory

Syntax: **C:\>DEL<FILE NAME>**

11. VER

It displays the version of DOS currently being used in the system.

Syntax: **C:\> VER**

12. PROMPT

Promt allows the user to se a new DOS prompt instead of usual **C:\> or A:\>** Syntax: **C;\>prompt pcc. Prompt pg** – this allows you to reset default prompt. **Prompt $d**(current date). **Prompt $t**(current time).

2.5.2 External commands

DOS commands for which specifications are not internally available in commands.com are called External commands. They reside in the disk in the form of executable program files. They will be loaded into primary memory only at the time of execution.

a) Command Syntax

For example: DIR d:pathname\filename.ext /p /w is the complete command for a disk directory. Note that only DIR is necessary. In some commands you may use wildcards. A wildcard, like the joker in a card deck, can stand for any character or group of characters.

- The **?**represents any single character: **FILE? = FILE1 or FILED** etc.
- The * represents any group of characters:
- ***.***= any file and extension

Use caution with wildcards. They can be dangerous with commands that do things like erasefiles. Also, in some cases a wildcard formulation can be misleading. The combination AP*EX.COM does not mean all files that start with AP and end with EX in their root name and with a COM extension. It means all files starting with AP and having an extension of COM. The EX is meaningless as it is ignored because of the asterisk.

Disk Directory

To see a listing of what is on a disk, issue the **DIR**ectory command.

DIR d:filename.ext /p /w

Three Simple Commands

VER: Shows the DOS version number on the video display. You are shown the one-digit version
and two-digit revision. For example MS-DOS Version 6.00

VOL d: Displays a volume label, if one exists. The label is a name you have given to the disk when it was formatted. It is used for identification purposes. (The serial number is put on the disk by the FORMAT utility.) Volume in drive C is HANDBOOK Volume Serial Number is 2C35-16F9

Date and Time : These two commands show and/or set the system date and time.

New computers have the clock built-in and do not require a program to load the time. If your clock battery fails, the default values will be 13-8-18 for the date and 00:00:00.00 for time. Now and again you will see files with a create date of 13/8/18; they were created on a system where the clock has failed and DOS has used its default value. For the DATE command you can enter the date as month/day/year with hyphens or slashes, i.e., 13/8/18 or 13-8-18 are acceptable dates. Do not enter the day of the week, even though it shows on the screen. The computer will calculate it for you.

The format for DATE is:

DATE <date> : On early computers the time setting required a 24-hour clock, i.e., any time after noon had to have 12 added to it, for example 3:00 pm had to be entered as 15:00. While the TIME command will still respond to this type of time, you may not also enter 3:00p and the computer is smart enough to know you mean 15:00.

The format for TIME is:

TIME <time> :

On most computers these commands will change the permanent clock settings as well as changing the date/time in DOS.

Disks straight out of the package need to be formatted, that is have tracks and sectors defined so DOS can find programs and data on the disk. The command syntax is below (only the most useful options are shown).

FORMAT d: /s /u :

where

- **d:** defines the disk that will be formatted
- **/s** puts the DOS system on disk to make it bootable
- **/u** specifies an unconditional format (can't unformat the disk)

An example: Problem: Format the disk in drive A: without UNFORMAT information.

Answer: The proper command is:

C>FORMAT A: /U

b) Checking the Disk

Now and again it is useful to check the integrity of the disk directory and file allocation table (FAT). The FAT is so important to the disk that there are two copies of it on each disk. The **CHKDSK** program does this for you. The basic format is:

CHKDSK d:filename.ext /f /v

Using the filename causes it to be checked for continuity (i.e., being stored on contiguous sectors on the disk for more efficient access).

/f tells DOS to automatically fix the FAT and other problems.

- **/v** is a verbose mode that shows progress as disk checking is taking place

c) Backing Up a Floppy Disk:

Floppy disks wear out after several hundred spin hours. Well before then you should have made a copy of the disk to preserve the integrity of its contents. You can, of course **FORMAT** and then **COPY *.*** to accomplish this. There is a quicker way however:

DISKCOPY d1: d2:

If you do not give drive specifications, the utility will ask for them. All information on the target disk will be destroyed, and **DISKCOPY** will format the target if it is found blank. Be careful, it's easy to destroy data by putting the disks in backwards!

Problem: Copy disk A: to B: Issue the proper command.
Answer: C:\>DISKCOPY A: B:

d) Erasing Files

Files you no longer need should be deleted from your disk to make room for more current files.
Use the **ERASE (DEL**ete) command for this:

ERASE d:FILENAME.ext
or
DELd:FILENAME.ext

Be careful, typographic errors in this command can bring disaster! You are allowed to delete all files on a disk with the wildcard *** (ERASE *.*)**, but DOS will question you. Recovery BEFORE writing anything else to disk is possible. An UNDELETE utility started shipping with DOS 5.0. Before that commercial utilities were available.

e) Renaming Files

For whatever reason, you may need to change the name of a file on your disk. Use this format:

RENAMEd:OLDNAME.extNEWNAME.ext
Wildcards are allowed, but can cause trouble if you are not careful.
The rename command will give you an error message if **NEWNAME** exists.

f) Copying Files

The **COPY** command is a very powerful command within DOS. With it you can create duplicates of individual files, join several files into one, and even use your computer like a simple typewriter by "copying" from the device named CON: to the device named PRN Copying one file to another (copies from filename1 to filename2):

COPY d1:FILENAME1.ext d2:filename2.ext/v /v option verifies the copy as it takes place. This adds confidence at the price of slower operation. There are other options not shown here. Wildcards are allowed. For example,

C:\>COPY ADDRS.LST B: Copies the single file **ADDRS.LST** from **C: to B:**.
C:\>COPY *.* B:/V Copies all files on **C:** to the disk in **B:** and verifies the information is it is being written.

C:\>COPY ADDRS.LST Yields an error message. Can't copy a file to itself.

C:\>COPY B:*.* Copies all files from drive B: to drive C:. (If a destination is not specified, the default drive and directory is used.)
Copy can also be used to concatenate (join) several files by using the following form:

COPY d1:FILENAME1.ext+d2:FILENAME2.ext+... d0:filename0.ext/v

The options are the same as the previous version of the copy command. All specified filenames (#1, #2, etc.) will be copied and joined into filename0. If no filename0 isspecified, the first source file named will be used. Wildcards are dangerous with this command.

Example:
Contents of **FILE1:** This is file number one
Contents of **FILE2:** This is file number two

C:\>COPY FILE1+FILE2 FILE3

Contents of **FILE3:** This is file number oneThis is file number two The **COPY** command can be used to create text files by copying from device **CON:** to a file.

The procedure is outlined in the text of the example below.

C:\>COPY CON: TEXTFILE:

This is the text to go into the text file being created. Each line is typed to the screen and it is being saved into a buffer for later transfer to the file TEXTFILE. Each line may be corrected as it is typed, but cannot be changed after it is terminated by the carriage return. Also, if you happen to type beyond column 80 on the screen, you cannot correct anything on the line above. Each line must be terminated by a carriage return (the enter key). You signal you are finished by typing a Control-Z, the symbol for end-of-file, followed by Return. ^Z .**1 File(s) copied**

XCOPY

For copying multiple files the **XCOPY** command can be a powerful ally. As its name implies, the command performs extended copies.
Its format (with only often-used options) is shown here:

XCOPY d1:PATH1 d2:path2 /a /m /s /v

Like the **COPY** command, **XCOPY** can take a single drive/path designator in which case files from that destination will be copied into the current directory.

Some options:

- **/A** Copy only files with archive bit set; do not reset archive bit.
- **/M** Copy only files with archive bit set; reset archive bit.
- **/S** Copy subdirectories as well unless they are empty.
- **/V** Verify copied files as they are written.

You can copy an entire hard disk to another disk with a single command:

```
XCOPY C: D: /S
```

The contents of drive C: will be copied to drive D: a file at a time, with the subdirectory structure intact. You can use the same technique to back up a hard disk to a removable disk.

Note the /M option above. When DOS writes a file to the disk it sets an archive bit in the disk directory to indicate the file has been somehow changed.

To proceed:

1) Make a full backup first. Use the **~ATTRIB~** command to set all archive bits to ON:

```
C:\>ATTRIB +A *.* /S
```

Use **XCOPY** to copy all files and directories, turning all archive bits OFF in the process (assumes
removable media is **G:**):

```
C:\>XCOPY C:\ G: /M /S
```

2) On a regular basis use the **XCOPY** command to perform an incremental backup:

```
C:\>XCOPY C:\ G: /M /S
```

The backup on drive **G:** will be an image of the file and directory structure on drive **C:**. The incremental backup makes certain the image is current. Periodically, in order to purge deleted files from the backup you should start over at #1 above and a clean backup disk.

Typing a File

Any text file saved in ASCII character format can be easily seen on your video display. Use the type command:

```
TYPE d:FILENAME.ext
```

All characters in the file will be displayed on the screen, including any control characters,

sometimes resulting in interesting displays. Any control-I characters found will be interpreted as a tab, and spaces will be added to get the cursor over to the next 8-character boundary; some output may appear as tables. Control-Z will cause output to stop. Attempting to **TYPEa .COM** or **.EXE** file will result in garbage on the screen, and should generally be avoided.

g) Subdirectory Introduction

As the name implies, a pathname is nothing more than a "path" that directs DOS to your particular file.

Pathnames

Assume the subdirectory structure (only directories are shown, not files)...

ROOT	LEVEL 1 SUBDIRECTORY	LEVER 2 SUBDIRECTORY
C:\	WORDPROC	BOOK MEMOS LETTERS

This series of subdirectories was set up to categorize various files developed by a wordprocessor.

Let's move in the structure:
\WORDPROC\LETTERS would be the pathname from the root to subdirectory LETTERS.

\WORDPROC Is the pathname from root to **WORDPROC.**

Note that each subdirectory in the path is separated by a backslash (\). The single backslash at The beginning of the pathname indicates the root. All pathnames must originate in either the current directory or root.

When DOS is booted, the root directory is automatically selected. To type a file named
MYMEMO.TXT in subdirectory MEMOS the command would be:

```
C:\> TYPE \WORDPROC\MEMOS\MYMEMO.TXT
```

\WORDPROC\MEMOS\ is the pathname that **DOS** would use to find the file **MYMEMO.TXT** and then show it on the screen.

Change Directory command

To change to the MEMOS subdirectory from the root the command would be:

```
C:\> CD \WORDPROC\MEMOS
```

Make and Remove Subdirectories
If you are going to have subdirectories, there must be a way to make them. The syntax for
the make directory command is:

```
MD d:pathname\DIRNAME
```

You can make a subdirectory IN any directory FROM any directory so long as you give the appropriate pathname. Usually, you will change to the directory you want the subdirectory to be in **MD DIRNAME** command. That way there is no mistake about what will happen. When you no longer need a directory you may remove it from the disk. The first thing you have to do is empty it of files and move out of it. Only then will you be able to remove it. The syntax for removing is:

RD d:pathname\DIRNAME

Note :You cannot remove the root directory

2.6 Summary

- Software is the part of the computer system which enables the hardware to operate. Computer Software can be divided into two major classifications
    - System Software
    - Application Software

- An operating system is a large collection of software, which manages resources of the computer system, such as memory, processor, and file system and input/output devices.

- It keeps track of the status of each resource and decides who will have a control over computer resources, for how long and when.

- Operating System is also called as Resource manager.

- **Functions of Operating System:** Even the simplest operating system in a minicomputer or mainframe performs a number of resource management tasks or functions. These functions include job management, batch processing, on-line processing, data management, virtual storage, and input/output management.

- **Time Sharing Operating System:** A time sharing system allows many users to share the computer resources simultaneously

- **Multi-user:** Multi-user operating systems allow two or more users to run programs at the same time

- **Multiprocessing:** Multiprocessing refers to a computer system's ability to support more than one process (program) at the same time

- **Multitasking:** Multitasking allows more than one program to run concurrently.

2.7 SAQ

1. Define Operating System?
2. List and explain different types of OS?
3. Explain Internal and External Commands?

Unit – III
Introduction to MS – Word

Structure :

3.0 Objective

Microsoft word is part of Microsoft's office suite of software. It is a word processor, which is world's leading word processing application. The primary objective is to enable students to create and edit document, save it to view or edit later and print document and many more features.

3.1 Introduction

Microsoft Word 2007 has a completely redesigned user interface. The standard menus along the top have been removed and replaced with a series of toolbars Microsoft calls "The Ribbon." The tool bars are changed using tabs at the top and try to automatically adjust themselves to the content you are working with. If you select an image it will automatically switch to the picture tools. The new layout seems to offer easier access to most of the features of Word allowing for more complex documents to be created quicker.

3.1.1 Features of Micro Soft Word (MS Word)

Microsoft Word is an industry-standard and world's most used word processing software. The program is used by educators, students, firstline workers, businesses, and enterprises. While it is very common for schools, companies, and offices to have the program installed on their computers; Microsoft Word is also built and designed for personal use, and aims to help users view, edit, create, and share beautiful documents.

Not only editing, viewing MS Word also supports many other features such as spell check, Mail merge, find and replace, inserting water marks, inserting your header and footer, hyperlink

3.1.2 MS Word controls

Controls are new feature of Microsoft Word 2007. They partially replace form fields. create customized templates, such as corporate forms, or letter or fax templates, that use your own theme and that use Content Control fields. The best way to see what these are is to open one of the new boilerplate templates in Word 2007 and have a look.

Use Content Control fields

1. In Microsoft Word, click **Office**, **New** to display the **New Document** dialog box.
2. From the **Templates** frame click **Installed Templates**. These are the templates supplied by default with Microsoft Word 2007.
3. Find and click on **Median Letter**, then click **Create**.
4. Look at the instructions surrounded by square brackets – these are called **Place Holders**. When you click on one of these, a box displays around it. When you type your own text in the brackets, the Place Holder disappears and just your own text remains.

Remove Content Control

1. Right click on the **Place Holder**.
2. Click **Remove Content Control**. You are returned to the document and there is no longer a Place Holder. (Note that this option is only available if the template designer made it available.)

Create Content Control fields

If you are creating templates for others to use, you may want to create content control fields like these. For this you will need the Developer tab.

1. Click the **Office** button, **Word Options** to display the **Word Options** dialog box.
2. With **Popular** highlighted, tick **Show Developer tab in the Ribbon** and click **OK**.
3. Back in your document, click on the **Developer** tab, and identify the **Controls** group.
4. Click **Design Mode**. This is a toggle.
5. Click the **Aa** button to insert a **Rich Text** content control.
6. Type any changes you want to make to the **Place Holder** text.

   You can type very detailed instructions here, formatted in any way you like. You can set an option that automatically removes all your content when the user starts typing in the control.
7. With your cursor in the field you have just created, click **Properties**, to display the **Content Control Properties** dialog box.

8. Complete the fields in the dialog box as required.
9. Click **OK** to save Properties.
10. Click **Design Mode** to toggle **Design Mode off** and see how the Content Control will display in a document.

3.1.3 Toolbars

Use toolbars in addition to or in place of menu bars. Toolbars can bemore efficient than menu bars because they are direct (always displayed instead of being displayed on mouse click), immediate (instead of requiring additional input) and contain the most frequently used commands (instead of a comprehensive list). In contrast to menu bars, toolbars don't have to be comprehensive or self-explanatory just quick, convenient, and efficient.

Some toolbars are customizable, allowing users to add or remove toolbars, change their size and location, and even change their contents. Some types of toolbars can be undocked, resulting in a palette window

Microsoft Word 2007 has a completely redesigned user interface. The standard menus along the top have been removed and replaced with a series of toolbars Microsoft calls “The Ribbon.” These tool bars are changed using tabs at the top and try to automatically adjust themselves to the content you are working with. If you select an image it will automatically switch to the picture tools. The new layout seems to offer easier access to most of the features of Word allowing for more complex documents to be created quicker.

3.1.4 Creating and saving document:

The first thing most people will need to relearn is where to go to open a document, create a new one, save your document and print. Clicking the Office logo at the top left of the screen will provide most of the items formerly found under the file menu including those listed above. Beside the logo you also will find a disk icon to save your document as well as the undo and redo buttons.

To create a new blank document:

1. Click the Microsoft Office button.
2. Select New. The New Document dialog box appears.
3. Select Blank document under the Blank and recent section. It will be highlighted by default.
4. Click Create. A new blank document appears in the Word window.

   Start a document from a template

To use a template as a starting point, do one of the following:

1. In Microsoft Office Word 2010, 2013 or 2016, click File, and then click New. In Microsoft Office Word 2007, click the Microsoft Office Button , and then click New.
2. Under Templates, do one of the following:
   - If you are using Microsoft Office Word 2007, click Installed Templates to select a template that is available on your computer. If you are using Microsoft Office Word 2010, look in Available Templates
   - In Microsoft Office Word 2007, click one of the links under Microsoft Office Online. In Microsoft Office Word 2010, click Office.com Templates. In Microsoft Office Word 2013 and 2016, all available installed templates are shown when you click File, and then click New and additional templates can be discovered by using the Search for online templates search box.
3. Double-click the template that you want.

To save a copy of your file, do the following

1. Click the Microsoft Office Button, and then click Save As, or press CTRL+S. In the File name box, enter a new name for the file.
2. Click Save.

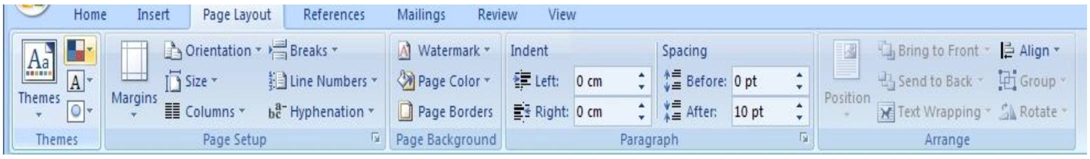
Note: If you don't see the Microsoft Office Button, click Save As on the File menu.

3.1.5 Opening a file (CRTL+N)

1. Click the Microsoft Office Button, and then click Open.
2. In the File of type list, click OpenDocument Text.
3. Click the file you want to open, and then click Open.

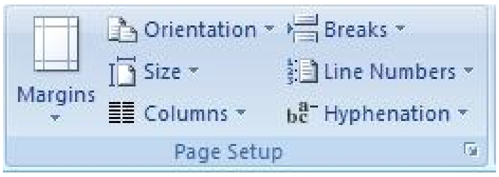
3.2. Page layout

The Page Layout Tab lets you control the look and feel of your document in Microsoft Word, a global design to your document by using one of the available themes and colour schemes. Changing the document orientation, page size, margins, indentation, line spacing and paragraph settings can be performed.

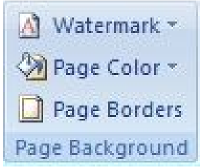


Indent Left: 0 cm
Indent Right: 0 cm
Spacing Before: 0 pt
Spacing After: 10 pt

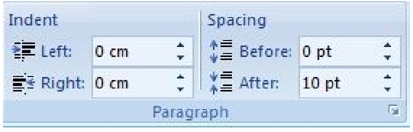
Themes: The themes section provides a quick way to format your document. By choosing a theme you will have a set colour scheme, font combinations, and effects. You can choose one of the provided themes, modify a provided them or create your own. Be aware that changing your theme after creating a document may require you to reformat some items as themes also include some layout options.



Page Setup: Page setup provides you with the tools to change margins, size, orientation, columns, breaks, line numbers and hyphenation in the document.

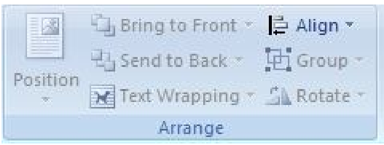


Page Background: The page background section allows you to change the background colour of the document, watermark and draft or confidential document or add borders to your document.



Left: 0 cm
Right: 0 cm
Before: 0 pt
After: 10 pt

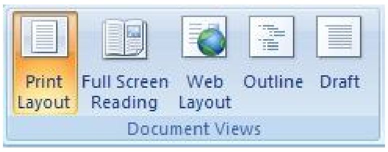
Paragraph: The paragraph section in the page layout tab allows changes to a paragraphs spacing and indentation.



Arrange: The arrange section is also found in the image toolbar when an image is selected. Here you can change an images position, how text moves around the image, the alignment, grouping and rotation or the image.

3.2.1 Print layout

Print layout is also known as reading layout. Print layout is the default view. Full screen view removes all but a couple of tools from the top of the screen and the rest of the screen is your document. Print lay provides use with a view of the document where we can see page breaks and other features such as the basic character and paragraph formatting. This is helpful when user want to see where text ends on a page or where paragraphs break over two pages. In one sentence we can say that print layout helps to view the document as it appears on the printed page. All types of layouts can be seen in the figure given below.



3.2.2 Web layout

Web layout will take away the empty space on either side of the document if there is any and fill the window as if it were a web page. Simulates the way a document will look when it is view as a Web page. Text and graphics appear the way they would in a Web browser.

3.2.3 Normal layout

The document views section switches user between print layout, full screen reading, web layout, outline and draft view. The above-mentioned document views web layout we have already gone through. Normal layout choice gives the largest area of window space that can fit into a document window. With this view, your document will automatically zoom to eliminate unneeded white space and fill up the area with as much text as possible. This should be the default view under which you do most of your typing.

3.2.4 Outline view

This presents the same document in outline form. In this view it is possible to collapse and expand different parts of the document. This makes it easier to work on these different parts. It is also easy to move blocks of text around the document should you decide that the order of things needs to be altered.

When you switch to **Outline View** a new menu bar appears that enables you manipulate the outline in different ways including, promoting and demoting headers, and expanding and contracting different parts of the document.

The document will only work as an outline provided the author has developed the document as an outline. This present document has been prepared as an outline, so, by switching to **Outline View** you will be able to see its various features.

The main benefits of **Outline View** are that it is very easy to move parts of the document around if you want to change the order of the argument. It is also very easy to make sure that formatting is consistent. And it is very easy to create **tables of contents**.

If you have a large document that has not been created as an outline, it is still possible to reconfigure it as an outline and then reap the benefits of outlines.

3.2.5 Setting tabs

Tabs are a paragraph-formatting feature used to align text. When you press the Tab key, Word inserts a tab character and moves the insertion point to the tab setting, called the tab stop. You can set custom tabs or use Word's default tab settings.

Tabs are set to distribute text evenly between the left and right margins. Word's default tabs are set every half-inch. These tabs are indicated at the bottom of the horizontal ruler by tiny tick marks. If you don't want to use the default tabs that are set at half-inch tabs, you have two choices. Change the distance between the default/existing tabs or create custom tabs. The four most common types of custom tabs are left-aligned, centered, right-aligned, and decimal-aligned. Custom tab settings are indicated on the horizontal ruler by tab markers.

There are two ways to set tabs. By either using the Tabs Dialog Box or using the ruler. Setting tabs by using the ruler is an easy, two-step process. Click the Tab Alignment button on the left of the ruler to choose the type of alignment and then click the position on the ruler to set the tab.

Adjusting Tab Settings : You can adjust tabs inserted in a document by using either the Tabs dialog box or the ruler.

3.2.6 Hyperlink

A **hyperlink** is a piece of **text** or an **image** in an electronic document that can connect readers to another portion of the document or a different webpage. In addition, a hyperlink can be used to connect to and open an email client window.

Have you noticed the **active links** on webpages that allow you to **jump from one page to another?** You just used one to open this lesson. These are called **hyperlinks**. You can use these in electronic versions of your Word documents, just like you do in webpages. In this lesson, you will learn the **basics of working** with hyperlinks, including how to insert and remove them in your Word document.

To insert a hyperlink:

- Select the **text or image** you want to make a **hyperlink**.
- Select the **Insert** tab.

- Click **Hyperlink** in the Links group. The Insert Hyperlink dialog box appears. If you selected text, the words will appear in the **Text to display:**field at the top.
- Type the address you want to link to in the **Address:** field.
- Click **OK**. The text or image you selected will now be a hyperlink.

To make an email address a hyperlink:

- Type the **email address**.
- Highlight the text, and select the **Insert** tab.
- Click **Hyperlink** in the Links group, and the Insert Hyperlink box appears.
- On the left of the box where it says **Link to**: click **Email Address**.
- The **Insert Hyperlink** box will change to the one shown above.
- Type the email address you want to connect to in the **Email Address** box, then click OK.

To remove a hyperlink:

- Select the **hyperlink** you want to deactivate.
- Select the **Insert** tab.
- Click **Hyperlink** in the Links group. The Edit Hyperlink dialog box appears
- Click **Remove Link**.

3.2.7 Book mark

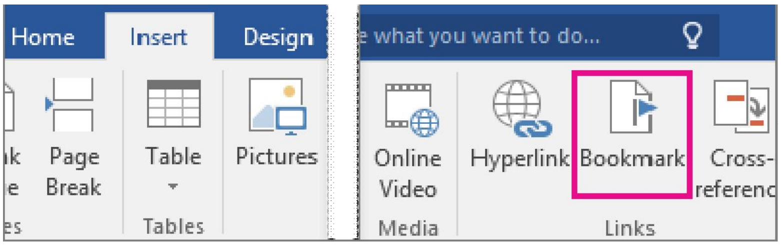
The bookmarks are the HTML elements that mark some locations in a document to enable the browsers to visit these locations immediately by activating so-called links. The bookmarks usually mark the headers of chapters and sections, the citations, the index entries etc.

A bookmark in Word works like a bookmark you might place in a book: it marks a place that you want to find again easily. You can enter as many bookmarks as you want in your document or Outlook message, and you can give each one a unique name so they're easy to identify.

To add a bookmark, you first mark the bookmark location in your document. After that, you can jump to the location or add links to it within your document or Outlook message. You can also delete bookmarks from a document or Outlook message.

Bookmark the location

1. Select text, a picture, or a place in your document where you want to insert a bookmark.
2. Click **Insert** > **Bookmark**.



3. Under **Bookmark name**, type a name and click **Add**.

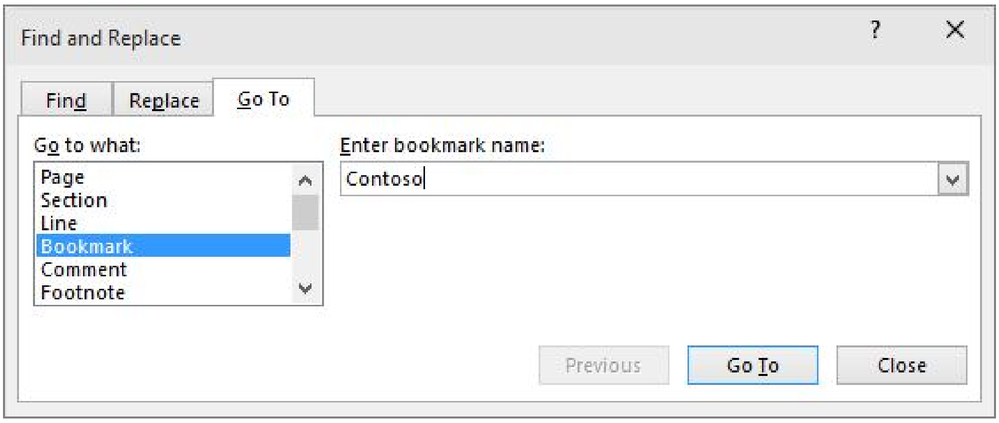
Note: Bookmark names need to begin with a letter. They can include both numbers and letters, but not spaces. If you need to separate words, you can use an underscore (_). For examplefirst_heading

Go to the bookmarked location

After creating your bookmarks, you can add links to them within your document or jump to them at any time.

Jump to a bookmark

Type Ctrl+G to open the **Go To** tab in the **Find and Replace** box. Under **Go to what**, click **Bookmark**. Enter or select the bookmark name, and then click **Go To**.



Go to what: Bookmark

Enter bookmark name: Contoso

Link to a bookmark

You can also add hyperlinks that will take you to a bookmarked location in the same document.

1. Select the text or object you want to use as a hyperlink.
2. Right-click and then click **Hyperlink** .
3. Under **Link to**, click **Place in This Document**.
4. In the list, select the heading or bookmark that you want to link to.
5. Click **OK**..

Delete a bookmark

1. Click **Insert** > **Bookmark**.
2. Click either **Name** or **Location** to sort the list of bookmarks in the document.
3. Click the name of the bookmark you want to delete, and then click **Delete**.
4. If you have inserted a hyperlink to the deleted bookmark, right-click the linked text and then click **Remove Hyperlink**.

3.3 Formatting Text

Formatting text in Microsoft Word involves tasks like bolding the text, italicising it, and changing the font and size. The commands to perform all of these formatting tasks are found on the Home tab in the Font group. Select your text and then click on the required formatting button to see the effects.

3.3.1 Setting font size

1. Selecting Text

The first step in formatting any amount of text or an entire document is selecting the portion of text that needs **formatting**. For formatting text, the following should be known

- Boldface a section
- Add italics
- Underline
- Change the colour
- Change the font
- Highlight
- Cut/copy and paste
- Add bullets

To do any of this, you must first select the text that the formatting will be applied to within the document. Selecting text is easy and done in three easy steps.

1. Move the cursor to the beginning of the text that you want to select.
2. Click and hold in the left button on your mouse.
3. Drag it over the text you want to select. It will highlight the selected text in **blue**. Simply release the mouse button when you are finished selecting text.

If you want to **select the entire document**, simply click on the '**Home'** tab and click '**Select'** on the far right. You will then select '**select all'.** This will highlight all your text for formatting. Also note that you can select objects or text with similar formatting as your selected text when clicking on '**Select**.'

Applying a Font

You can apply a font to either an entire document or to certain portions of a document. The choice is yours. To apply a font, click on the **Home** tab. You'll see the toolbars, or groups.

The type of font you're currently using will appear. In the picture above, the current type of font is Calibri. The notation of (body) after Calibri lets the user know that it's the font used in the body of text, not in headers, footnotes, etc. You can click on the arrow in that window to display other types of font available to use. To change the type of font within a document or for the entire document, simply select the text to be formatted, then select your new font. It's really that easy.

Note: You can also do all fonts formatting by selecting the arrow key beside the word Font, located directly above the document body. By clicking on this arrow, you can also alter character spacing.

Changing Font Size

Next to the type of font under the 'Home' tab is a box that states the size of font currently being used. Using the picture above, the font size is 12. Once again, you can use the down arrow key to either enlarge or reduce the font size. To change the size of the font, simply select the text that you want to change, then select the new font size. You can change an entire document or just a portion. The choice is yours. You can also use the two uppercase A's beside the font size box to change the size of fonts you're currently using. Pushing the larger A will increase your font size by one for each time it's clicked. On the other hand, clicking the smaller A will decrease the font size by one for each time it's clicked.

Font Attributes

As we discussed earlier, you may want to select text to either boldface, italicize, or underline a section of text. The boldface command in MS Word is represented by an uppercase, boldfaced B. Italics are represented by an uppercase, italicized 'I', and underline by an uppercase U with a line under it. These button's are located directly below the font type window. To add *italic*, **boldfaced** or underlining to any portion of a text within a document, select the desired text, then click the appropriate button (ctrl +**B** for boldfaced, ctrl + **I** for italic, or ctrl + **U** for underline.)

Next to the commands for boldfaced, italics, and underline, you'll see an 'abc' with a line through it. (~~abc~~) This is for strikethrough. When clicking on this button, it will put a line through any text you have selected.

Next to the strikethrough command, you'll see 'X_2'. This simply puts smaller text below the text line of the normal text. This command was used when typing X_2 into this course. X^2 is just the opposite, putting smaller text above the text line. This makes creating footnotes easier than it had been with previous versions of Word.

The uppercase A and lowercase A that are situated side by side allow you to change the case of selected text. You can change from lowercase to uppercase or vice versa , toggle tHeCaSe, or select other common capitalizations.

3.3.2 Font Color

Changing the font color is as simple as changing the font. Look for the uppercase A with a black bar under it as pictured below. It's just to the right of the Aa that allows you to capitalize portions of text. Select your text, then click on the button to choose the colour you want to apply to the selected text. You can also highlight selected text. To the left of the font colour button, you'll see a lowercase 'ab' with a marker near it. It is highlighted in yellow. Simply select the portion of text you want to highlight, click the button, and select the colour of highlight that you want to apply.

3.4 formatting paragraph

When you format paragraphs within a document, you can change the look of the entire document. Formatting a paragraph allows you to change line spacing, indent lines, and alter all of your paragraphs or just a few so that they look the way you want them. Many of the tools you can use to format your paragraphs will be located in the '**Page Layout**' tab of the Ribbon in the '**Paragraph**' group.

3.4.1 Aligning text

As with all word processing programs, you can either left, right, center or justify your text. You can either do one of these things to a portion of text or to the entire document.
For example the below text which is aligned to the left and center this is text that is aligned to the right.

This text is centered.

Justified text: it is aligned between the left and right margins, adding space between letters if necessary. It gives the document a clean look. The buttons or commands for aligning text are located under the 'Home' tab above the 'Paragraph'. The Home tab consists of the left alignment button beside it is the center, then right, and then justify. To use these, simply select the text you want to align, and click on the appropriate button.

3.4.2 Aligning line spacing

Line spacing Adjusting line spacing changes the amount of space between lines of text. Line spacing can be single, 1.5, double, multiple or you can choose your own size. Adjusting line spacing

- Select the text that you want to adjust the line spacing on. If you want to change the whole document, choose Select then Select All from the Editing group
- Click the arrow button to the right of the Paragraph group to open the Paragraph dialog box
- Select the Indents and Spacing tab
- Click on the down arrow below line spacing. Select a new size
- Click OK.

3.5 Header and footer

Header and footer can make your document look professional and polished by utilizing the header and footer sections. The header is a section of the document that appears in the top margin, while the footer is a section of the document that appears in the bottom margin. Headers and footers generally contain information such as page number, date, and document name.

To insert a header or footer:

- Select the Insert tab.
- Click either the Header or Footer command. A menu appears with a list of built-in options you can use.
- Left-click one of the built-in options, and it will appear in the document.

OR

- Left-click Blank to select it.
- The Design tab with Header and Footer tools is active.
- Type information into the header or footer.

To insert the date or time into a header or footer:

- With the header or footer section active, click the Date & Time command.
- Select a date format in the dialog box that appears.
- Click OK. The date and time now appear in the document.

Other header and footer options

There are many other header and footer options you can use to design these sections of your document. From the Header and Footer Tools Design tab, you can see all of your design options.

3.5.1 Page numbering

1. On the Insert tab, in the "Header & Footer" group, click Page Number.
2. Click Top of Page, Bottom of Page, or Page Margins, depending on where you want page numbers to appear in your document.
3. Choose a page numbering design from the gallery of designs.
4. On the Design tab, in the "Options" group, choose Different First Page.

Page numbering in header and footer can be set for alignment similar to that of a paragraph like center/left/right based on the requirement which is available in the header and footer options.

3.6 Tables

A table is made up of rows and columns. The intersection of a row and column is called a cell. Tables are often used to organize and present information, but they have a variety of uses as well. You can use tables to align numbers and create interesting page layouts.

3.6.1. Creating Table

Table can be easily made in MS Word. The following is the procedure to create table.

Click the New File icon to create a new document if necessary. To Create a Table:

- Click the Insert tab on the Ribbon
- Click the Table button
- Select Insert Table
- Click the arrows to select the desired number of columns
- Click the arrows to select the desired number of rows
- Click OK

Formatting a Table: Using Microsoft Word you are able to format a table by changing table lines and colors, shading tables, adjusting row and column size as well as alignment.

To Shade the Cells/Table:

- Position your cursor in the table
- Click the Layout tab on the Ribbon
- Click the Select button in the Table group
- Choose the item you wish to format
- Click the Design tab on the Ribbon
- Click the Shading button
- Select a shading color

To Adjust the Line Style of a Cell/Table:

- Position your cursor in the table
- Click the Layout tab on the Ribbon
- Click the Select button in the Table group
- Choose the item you wish to format
- Click the Design tab on the Ribbon
- Select the Line Style drop-Down Arrow
- Select a style
- Click your mouse in the table where you want to apply the line style to
- Press ESC key on the keyboard when you are finished applying the style.

3.6.2 Inserting, deleting and merging rows and columns

a) To Insert a Row:

- Position the cursor in the table where you would like to insert a row
- Select the Layout tab on the Ribbon
- Click either the Insert Row Above or the Insert Row Below button in the Rows & Columns group

b) **To Insert a Column**

- Position the cursor in the table where you would like to insert a column
- Select the Layout tab on the Ribbon
- Click either the Insert Columns to Left button or the Insert Columns to Right button in the Rows & Columns group

c) To Delete a Row

- Position your cursor in the row that you would like to delete
- Select the Layout tab on the Ribbon
- Click the Delete button in the Rows & Column group
- Select Delete Rows

d) To Delete a Column

- Position your cursor in the column that you would like to delete
- Select the Layout tab on the Ribbon
- Click the Delete button in the Rows & Column group
- Select Delete Columns

e) To Merge Cells in a Table

- Select the cells that you would like to merge in the table
- Click on the Layout tab on the ribbon
- Click the Merge Cells button in the Merge group

3.6.3 Deleting table

Keep in mind that deleting the table deletes all its contents as well.

- Click the mouse inside the table. The Table Tools appear.
- If the Layout tab is not displayed, click to display it.
- Choose Delete—Table. This command is located in the Rows & Columns group on the Layout tab.

3.7 Mail merge

Mail merge is a tool which allows you to create form letters, mailing labels and envelopes by linking a main document to a set of data or data source. It is the process of combining a list of data with a template. The mail merge process involves the following:

The Main Document: Contains the text and graphics that are the same for each version of the merged document.

Data Source: A file that contains the information to be merged into a document. For example, the names and addresses of the recipients of a letter.

Mail Merge– Form Letters

- Open Microsoft Word. Type the letter with all needed text and formatting leaving areas for the data source data blank (ex name, address…)
- Click the Mailings tab
- Click Start Mail Merge
- Click Step by Step Mail Merge Wizard

The Mail Merge task pane appears on the right of your screen. There are 6 steps involved in Mail Merge.

Step 1

- Click Letters for the document type
- Click Next: Starting document

Step 2

- Click Use the current document under Select starting document
- Click Next: Select recipients

Step 3

The recipients can come from either an existing Excel file, or Access table or you can create a new list.

a) If Using an Existing List:

- Click Use an existing list under Select recipients
- Click Browse
- Select the file
- Click Open Mail Merge Recipients opens showing the names and addresses from your file
- Click OK
- Click Next: Write your letter

b)To Type a New List:

- Click Type a new list under Select recipients

- Click Create
- Click Customize Columns to modify the list of fields
- Delete any unnecessary field names and/or add new ones
- Click OK
- Type records here hitting TAB to advance to the next field and to continue adding new records
- Click OK
- Click Save

The recipients list will be saved as a separate file as a Microsoft Access file type. It is saved in the My Data Sources folder. It is recommended to save the file in this folder.

- Click Next: Write your letter

Step 4

- Click the location in your document where the data from the mail merge fields need to be inserted
- Click Address block…
- Select the address elements you want included
- Click OK

The field name will look like this: <> The Address block will insert the following fields including any necessary punctuation: First Name, Last Name, Company, Address 1, Address 2, City, State, Postal Code. If your fields do not match the ones listed above or you have additional fields, click More items…

- Click on the field from the list
- Click Insert
- Click Close The field name will look like this - «First_Name»
- Repeat this step until all fields have been inserted. Remember to put spaces and punctuation where needed.
- Click Next: Preview your letters

Step 5

Here is where you can preview the first page with the fields filled in. Click Next: Complete the merge

Step 6

- To Complete the Merge:
- Click Print to send directly to the printer
- Click Edit individual letters to create a new file Remember to save your document as you go. The next time you open your document and click on Step by Step Mail Merge, the data file will be attached

Mail Merge – Labels

- Open a new blank document
- Click the Mailings tab
- Click Start Mail Merge
- Click Step by Step Mail Merge Wizard

Step 1

- Click Labels for the document type
- Click Next: Starting document

Step 2

- Click Label options under Change document layout
- Choose the label style you are using
- Click OK
- Click Next: Select recipients 6

Step 3

- Click Use an existing list under Select recipients
- Click Browse
- Select the file that contains the names and addresses
- Click OK
- Click Next: Arrange your labels

Step 4

- Click in the first label box and click on either Address block or More items to insert the data fields
- Click Update all labels to include the fields on all labels
- Click Next: Preview your labels Step 5 Here is where you can preview the labels. Click Next: Complete the merge Step 6 Click Print to send directly to the printer OR Click Edit individual labels to create a new file

3.Summary

- Microsoft Word 2007 has a completely redesigned user interface. Microsoft Word offers several features to ease document creation and editing, including:

- WYSIWYG (what-you-see-is-what-you-get) display: It ensures that everything displayed on screen appears the same way when printed or moved to another format or program.
- Spell check: Word features a built-in dictionary for spell checking; misspelled words are marked with a red squiggly underline. Sometimes, Word auto-corrects an obviously misspelled word or phrase.
- Text-level features such as bold, underline, italic and strike-through
- Page-level features such as indentation, paragraphing and justification
- External support: Word is compatible with many other programs, the most common being the other members of the Office suite.
- The default file format was .doc prior to the Microsoft Word 2007 version; in 2007, .docx became the default file.

3.8 Summary

- Microsoft Word is an industry-standard and world's most used word processing software. The program is used by educators, students, firstline workers, businesses, and enterprises.

- While it is very common for schools, companies, and offices to have the program installed on their computers

- Microsoft Word is also built and designed for personal use, and aims to help users view, edit, create, and share beautiful documents

- **To create a new blank document:**

    - Click the Microsoft Office button.
    - Select New. The New Document dialog box appears.
    - Select Blank document under the Blank and recent section. It will be highlighted by default.
    - Click Create. A new blank document appears in the Word window.
    - Start a document from a template

- A **hyperlink** is a piece of **text** or an **image** in an electronic document that can connect readers to another portion of the document or a different webpage.
- The bookmarks are the HTML elements that mark some locations in a document to enable the browsers to visit these locations immediately by activating so-called links. The bookmarks usually mark the headers of chapters and sections, the citations, the index entries etc.

- Formatting text in Microsoft Word involves tasks like bolding the text, italicising it, and changing the font and size. The commands to perform all of these formatting tasks are found on the Home tab in the Font group.

- Header and footer can make your document look professional and polished by utilizing the header and footer sections. The header is a section of the document that appears in the top margin, while the footer is a section of the document that appears in the bottom margin

- A table is made up of rows and columns

- **Mail Merge– Form Letters :**Open Microsoft Word. Type the letter with all needed text and formatting leaving areas for the data source data blank (ex name, address…)

3.9 SAQ

1. What is MS-Word?
2. List characteristics of MS-Word?
3. Explain what is Text Wrapping?
4. Explain what is WYSISYG?
5. Briefly explain about page set up?

Unit – IV
Introduction to MS – Power Point

Structure

4.0 Objective

The main objective of this MS Power Point to make the student perfect in creating presentation, deleting unnecessary slides, slide numbering, inserting graph, picture, sound and video organization chart. Even assigning transition and timing for slide. How to print slide.

4.1 Introduction

PowerPoint is the presentation program included in the Microsoft Office suite. With PowerPoint, you can create engaging presentations that can be presented in person, online, or as self-running slideshows. Your presentations can include graphics, audio and video clips, and links to external websites. PowerPoint 2007 has a different interface from previous versions of the program.

4.1.1 Features of MS Power Point

PowerPoint is a highly innovative and versatile program that can ensure user a successful communication whether you're presenting in front of potential investors, a lecture theatre or simply in front of your colleagues. Below are five features you should be using. They will improve your presentation skills and allow you to communicate your message successfully.

- Adding smart art
- Inserting shapes
- Inserting an image
- Slide transitions
- Adding animation

4.1.2 Creating presentation

To create a new presentation with PowerPoint 2007:

1. Launch PowerPoint 2007. The program will open up with a new blank presentation already created for you.

2. To save the presentation, click on the new Office button and choose Save (or click on the Save button in the Quick Access Toolbar).

3. Enter a file name for your new presentation and select a save location.

4. Click on Save. PowerPoint 2007 uses a different file format than previous versions of the program. However, you can save a copy.

PowerPoint 2007 presentation that is backward-compatible with older versions of the program. To do this, click on the Office button and select Save As > PowerPoint 97-2003 Presentation. You should save often as your work on your presentation to avoid losing your changes. A quick way to save changes as you work is to use the keyboard shortcut Control + S. If you want to save a copy of your presentation with a different name, click on the Office button, choose Save As > PowerPoint Presentation and enter a different name in the File name field at the bottom of the Save As window.

Another way to create a new presentation: By clicking on the Office button and choosing New. This will open the New Presentation window, where you will have the following options on the left side:

- Blank and recent: click on Blank Presentation on the right to create a blank presentation with no formatting applied to it.
- Installed Templates: these are the built-in presentation designs included with PowerPoint 2007.
- Installed Themes: similar to templates, but also include formatting such as colors, fonts and effects to give your presentations a consistent look.
- Microsoft Office Online: these are templates you can download from the Microsoft website if you have an internet connection. Once you've chosen a template or theme, click on Create or Download. The formatting options included in the template or theme will be applied to the new presentation created by PowerPoint 2007

4.1.3 Inserting and deleting a slide

Working with slides: To add a slide to your presentation, open the Home tab in the Ribbon and click on the New Slide button. This will add a new slide with the same layout as the previous slide. To change the layout of a slide after you've added it to your presentation, click on Layout in the Home tab and choose a different layout from the menu. To duplicate an existing slide, select the slide in the Slide Pane, right-click on it and choose Duplicate Slide.

To move a slide so it appears at a different point in the presentation, use drag and drop to move the slide's thumbnail in the Slides Pane (on the left side of the window) until it is in the desired location. If you have a long presentation with many slides, the Slide Sorter may provide a better view while you reorder your slides.

Another way of adding slide to the presentation

How to Add Slides to a Presentation

1. Click the New Slide button on the Formatting toolbar.

2. From the list of Slide Layouts, select the layout you want to apply to the new slide.

3. You may now begin adding content using the placeholders in the layout.

How to Change the Layout for any Slide

PowerPoint will try to guess what layout you want to use for new slides that are added to the presentation. If you want a different layout for the slide you can quickly change the layout for any slide.

1. Display the slide that you want to change in the Slide Pane (work area in the center of the window).

2. Choose Format > Slide Layout to display the Slide Layout task pane.

3. Click on the layout you want to apply to the slide.

4. PowerPoint will attempt to fit existing content into the new layout, but you will probably have to make additional changes.

How to Add Slides in the Outline Tab

Create new slides while working in the Outline tab. By default the Outlining toolbar should display, but if it does not, select View > Toolbars > Outlining.

1. Display the Outline tab by clicking on Outline in the pane on the left.

2. Place the cursor at the end of the text in the slide you wish the new slide to follow.

3. Click the New Slide button to insert a new slide.

To delete a slide, press the Delete key on your keyboard with the slide selected in the Slide Pane. To hide a slide, right-click on it in the Slide Pane and select Hide Slide.

The hidden slide will be indicated by a square around the slide number in the Slides Pane. To unhide the slide, right-click and select Hide Slide one more time.

Adding Content to Slides: The fastest way to add content in PowerPoint is by selecting a slide layout and entering your content into one of the text placeholders. For example, if you select the Title Slide layout, you can enter a title for your slide by clicking on the placeholder text at the top of the Slide Canvas and entering your own text.

You can also add content to a slide by copying it from another document the following step should be implemented

- Select the content you wish to copy in the other document and copy it to the clipboard. In most programs this is done by choosing Edit > Copy (if you are copying from a Microsoft Office document, click on the Copy button in the
- Clipboard group of the Home tab). Place your cursor inside the text placeholder where you want to paste the copied content.
- Click on the Paste button in the Clipboard group of the Home tab. Many slide layouts also includes bulleted lists. To add text to a bulleted list, click the placeholder text next to the first bullet and begin adding the desired text. A new item will be added to the list each time you press the Enter key on your keyboard. To go to the next line without adding another bullet, hold down the Shift key while pressing Enter.

Many slide layouts also include bulleted lists. To add text to a bulleted list, click the placeholder text next to the first bullet and begin adding the desired text. A new item will be added to the list each time you press the Enter key on your keyboard. To go to the next line without adding another bullet, hold down the Shift key while pressing Enter.

4.1.5 Saving Power Point

Office button includes a lot of options that used to be found under the file menu, such as save and print. Once new presentation with Power Point 2007 is launched by open up with a new blank presentation/already created by the user, the presentation should be saved or save as action should be taken place. The following are the steps for saving a presentation

1. To save the presentation, click on the new Office button and choose Save (or click on the Save button in the Quick Access Toolbar).
2. Enter a file name for your new presentation and select a save location.
3. Click on Save.

PowerPoint 2007 uses a different file format than previous versions of the program. However, you can save a copy of your PowerPoint 2007 presentation that is backward-compatible with older versions of the program. To do this, click on the Office button and select Save As > PowerPoint 97-2003 Presentation. You should save often as your work on your presentation to avoid losing your changes. A quick way to save changes as you work is to use the keyboard shortcut Control + S. If you want to save a copy of your presentation with a different name, click on the Office button, choose Save As > PowerPoint Presentation and enter a different name in the File name field at the bottom of the Save As window.

4.2 Inserting objects in Presentation

Slides in a PowerPoint 2007 presentation are filled with objects, such as text, clip art, charts, and graphs. Most objects on your PowerPoint2007 slides are text objects, which let you type text on your slides. ... Every slide has a slide layout that consists of one or more placeholders.

4.2.1 Inserting graph, picture, sound and chart

Inserting graph/chart

A chart is the graphical representation of numerical data. The following are the steps should be followed while inserting a chart or graph

- To insert a chart, click the Insert Chart icon in the content placeholder. There are different types of chart such as pie, line, column, bar, area and so on.

- The Insert Chart dialog box opens from which you can choose the type of chart you need.
- When you insert a chart object into PowerPoint, you are embedding it • An embedded object is an object whose data source can be opened for editing purposes. Changes you make to an embedded object in PowerPoint do not affect the data source

Inserting image/picture/clipart

Effective visuals emphasize the key content points in a presentation. PowerPoint provides a selection of professionally designed pictures, or clip art, that you can use in your presentations. These clip art images include many different themes such as animals, people, buildings, food, holidays, business, and more.

How to Insert a Clip Art Image

6. Move to the slide on which you want to place clip art.
7. Apply a Slide Layout that includes a content or clip art placeholder.
8. Open the Select a Picture dialog box by:

• Clicking on the Clip Art button on the content placeholder OR

• Double-clicking on the clip art placeholder

1. In the Search box, type a word or phrase that describes the clip you want.
2. Click Search. PowerPoint displays the search results in the Select Picture List.
3. Click on the clip art image you want and click OK.

How to Resize a Clip Art Image: Once you have added a clip art object to your slide, you can resize it to make it fit better into your presentation.

1. Click on the Clip Art object to select it.

2. Put the arrow on one of the resize handles at the corner of the picture until the cursor changes to a double-headed arrow.

3. Depress the mouse button and drag the handle toward or away from the center to make the image larger or smaller. The corner handles resize the image proportionally and the handles on the sides of the image increase or decrease the height or width of the image. When you release the mouse button, the object appears in its new size.

Insert and Style a Picture

In PowerPoint, a picture is artwork created in another program and inserted into a PowerPoint

- digital photograph
- a piece of line art or clip art
- other artwork

There are 17 types of picture formats that you can insert into PowerPoint • Examples include:

- JPEG
- BMP
- TIFF

To insert a picture, click the Picture button in the Illustrations group or

- Click Picture icon in a content placeholder
- Look for a picture in the Insert Picture dialog box, then click Insert

Crop a picture to remove unwanted areas

The Crop button is in the Size group of the Format tab

- The cropped portion of a picture, when you save unless you change the setting in the Compression Settings dialog box

Inserting sound in Power Point

Adding multimedia such as sound and video to your presentation can be a great way to catch and focus your audience's attention when used in moderation. PowerPoint 2007 provides some videos and sound clips that are free to use in the Clip Manager. Using clips from other sources, such as the Web, a CD of sound effects, or sounds you create yourself.

Inserting Sounds

Inserting sounds using the Clip Organizer

1. Select the slide to which you wish to add a sound.
2. In the Insert tab, under the Media Clips group, click the arrow on the 🔊 Sound button. Choose Sound From Clip Organizer... to bring up the Clip Art task pane.
3. Use the Search Box to search by keyword and click Go to search.
4. Click the sound to insert it, or hover your mouse over it first, click on the menu arrow that appears, and choose Insert.
5. A pop-up box will appear asking you how you would like the sound to start in the presentation. Choose to either have it play automatically or to have it play only when clicked.
6. A sound icon 🔊 appears on your slide.

Inserting sounds from a file

1. Select the slide to which you wish to add a sound.
2. In the Insert tab**,** under the Media Clips group, click the arrow on the 🔊 Sound button. Choose Sound From File... to bring up the Insert Sound dialog box.
3. Find the sound file on your computer using the "Look In" drop-down. Select the file and click Ok**.**
4. A pop-up box will appear asking you how you would like the sound to start in the presentation. Choose to either have it play automatically or to have it play only when clicked**.**
5. A sound icon 🔊 appears on your slide.

Recording your own sounds

PowerPoint 2007 offers different tools for recording narration.

1. Select the slide to which you wish to add a sound.

In the Insert tab, under the Media Clips group, click the arrow on the 🔊 Sound button. Choose Record Sound... to bring up the Record Sound dialog box.

1. In the *Name* box, type an appropriate name for your sound.
2. When you are ready click RECORD ● to record the sound.
3. When you are finished recording press STOP ■ .
4. To hear your sound click PLAY ▶.
5. If you are not satisfied with your recording, click Cancel and start again. If you are happy with the recording, click OK.
6. A sound icon 🔊 appears on your slide.

4.3 Animation

Already we have seen that MS PowerPoint, a presentation tool was used to create effective and eye-catching presentations in a matter of minutes. It gives you a step-by-step creation of a simple presentation, besides introducing you to the various features available in PowerPoint. It also details the way in which you could make your presentation attractive by changing the font (type, size and colour), adding pictures, diagrams and animation. Here now we study how animation makes our presentation an eye-catching one.

Transitions

To add a transition to a slide:

1. Choose the slide in the Slides Pane.

2. Open the Animations tab of the Ribbon.

3. Hover over the transition icons in the Transition to This Slide group to see a preview of each transition in the Slide Canvas. When you find one you like, click on the icon for that transition.

4. Use the pulldown menus to the right of the transition icons to add sound and change the transition speed.

5. Click on “Apply To All” if you wish to apply the currently selected transition to all slides in your presentation.

To advance the current slide automatically when you present, check the box next to “Automatically After” and enter the number of seconds in the field to the right.

Animation

To animate any object in the Slide Canvas:

1. Select the object you wish to animate.

2. Select an animation from the Animate pulldown menu in the Animations tab of the Ribbon. You will see a preview of the animation in the Slide Canvas.

3. If you wish to create more complex animations, choose Custom Animation at the bottom of the Animate pulldown menu. This will open the Custom Animation pane on the right side of the window.

4. Click on Add Effect and choose the type of animation you wish to add (Entrance, Emphasis, Exit, or the new Motion Path animation). Clicking on each animation type will open a submenu with the animations available under that category.

5. Once you’ve added an animation, use the pulldown menus under Modify to set when the animation will start (On Click, With Previous or After Previous) and the animation speed. Depending on the animation type you select, you may see additional pulldown menus for changing the animation direction or path.

6. Use the arrows under the animations list to change the order in which the animations will be executed when you play the slide. To remove an animation, select it from the list and click on the Remove button at the top of the pane.

7. Click on Play to preview your animation in the Slide Canvas.

Motion paths:

Motion paths allows you to select a path that will apply to the selected object or text. Various options are available and are detailed further in this unit. Motion paths is the ability to animate an object so it moves along a path. To add a motion path animation:

1. Select the object you wish to animate.

2. Choose Custom Animation from the Animate pulldown menu of the Animations tab. The Custom Animation task pane will open on the right side of the window.

3. Click on Add Effect and choose Motion Paths.

4. Select a path from the list or select Draw Motion path to draw the path yourself.

5. Each path will have a start point and an end point.

Selecting the desired effect: By applying one or more of these to see the effect on the object, item, or text and choose the one that appeals the most.

Effects available under each of the categories are shown below.

Entrance

(1) Blinds

(2) Box

(3) Checkerboard

(4) Diamond

(5) Fly In

Emphasis

(1) Change Font

(2) Change Font Size

(3) Change Font Style

(4) Grow/Shrink

(5) Spin

Exit

(1) Blinds

(2) Box

(3) Checkered box

(4) Diamond

(5) Fly In

Motion Paths

(1) Diagonal Down Right

(2) Diagonal Up Right

(3) Down

(4) Left

(5) Right

(6) Up

4.4 Finalizing presentation

You can modify the duration of a transition, modify the transition effect, and even specify a sound to play during a transition. Also, you can specify the time to spend on a slide before advancing to the next.

4.4.1 Assigning transition and timing

If no timing is selected, slides advance when you click the mouse.

1. Select the slide that you want to set the timing for.
2. On the Transitions tab, in the Timing group, under Advance Slide, do one of the following: To make the slide advance to the next slide when you click the mouse, select the On-Mouse Click check box.

4.5 Printing

In PowerPoint, you can designate the number of slides to print on a single page.

1. Go to the File menu and choose Print.
2. In the bottom left of the Print window that opens, you will see a pull-down menu under the words Print What. Choose Handouts.
3. The default setting is 6 slides per page.

4.7 Summary

In this unit, you have studied that a PowerPoint presentation consists of individual slides arranged in a sequential manner. New slides can be inserted, existing slides can be deleted and slides can be re-ordered within a presentation. Presentations can then be saved in the default directory (which can be changed, if required) with a **.ppt** extension.

You can also customize your presentation and control the appearance by adding pictures, objects and animation to it thereby giving it an attractive look.

4.8 SAQ

1. Write short notes on commonly used slides in Power Point?
2. Write short notes on how to implement animation in Power Point?
3. How to insert graphs in Power Point?
4. Discuss the steps involved in inserting sound and video in Power Point?
5. Discuss how to create presentation?

Unit – V
Introduction to MS Excel

Structure

5.0 Objective

students should be able to

- identify the MS Excel screen layout and its components, and
- identify various menus and their functions.
- know the MS Excel Standard Toolbar and the Formatting Toolbar and its components, and
- learn various functions and formulas of MS Excel.

5.1 Introduction

MS Excel is a powerful spreadsheet program through which you can create tables of any kind. These tables may contain numeric values, text data or formulas. You can use Excel to automate simple tasks like giving lists or to create sophisticated analysis tables for calculating the future trends in the stock market

5.2 Excel features

Conditional Formatting: You can format cells/data according to condition and can get glimpse of entire data within seconds.

VBA, Macros & automation: This would make computer work for you. You can automate mundane job for you.

Pivot Tables: Worried about how you'll interpret data. Let excel do it for you. This is to must know feature of excel.

Lookup Formulas: This formula is deadly. Combination of it with match, index can help in doing wonders.

PowerPivot: It helps in querying the huge data effectively. Has been introduced in recent version.

Add Header and Footer

MS Excel allows us to keep header and footer in out spreadsheet document.

Find and Replace Command

MS Excel allows us to find the needed data (text and number) in the workbook and also replace the existing data with a new one.

Password protection

It allows user to protect their workbooks by using password from unauthorized access to their information.

Data filtering

Filtering is a quick and easy way to find and work with a subset of data in a range. A filtered range displays only the rows that meet the criteria you specify for a column. MS Excel provides two commands for filtering ranges:

- Auto filter: which includes filter by selection, for simple criteria
- Advanced filter: for more complex criteria

Data sorting

Data sorting is the process of arranging data in some logical order. MS Excel allows us to sort data either in ascending or descending order.

Built in formulae

MS Excel has got many built – in formulae for sum, average, minimum, etc. We can use those formulae as per our needs.

Automatically edits the result

MS Excel automatically edits the result if any changes are made in any of the cell.

5.3 Worksheets

A Work sheet, in the word's original meaning, is a sheet of paper on which one performs work. Microsoft Excel, a popular spreadsheet program, refers to a single spreadsheet as a worksheet, and it refers to a collection of worksheets as a workbook.

5.3.1 Creating, Inserting and Deleting Worksheet

a) Creating new worksheet: To create new worksheet following steps should be implemented

- Click the file tab.
- Click new
- Under templates, click New from existing.
- In the New from Existing Workbook dialog box.
- Click the workbook, and then click Create New.

b) Inserting and deleting worksheet

Inserting and deleting worksheets in Excel is a critical skill to possess when working within a workbook in Excel.

To insert a single new worksheet at the end of the current listing of worksheets, click the "New Sheet" button at the right end of the spreadsheet name tabs. Alternatively, you can also click the "Home" tab in the Ribbon. Then click the "Insert" drop-down button within the "Cells" button group. Then select the "Insert Sheet" command from the drop-down menu. Alternatively, you could press the keyboard shortcut of "Shift"+"F11" on your keyboard to automatically insert a new worksheet. If using these last two methods, however, note that the new worksheets will always be inserted to the left of the worksheet name that appears in bold when the sheet is inserted.

Deleting old worksheets that you no longer use can reduce worksheet clutter in workbooks. You can easily delete worksheets in Excel. Simply select the worksheet name tabs of the sheet or the sheets that you want to delete. Then click the "Delete" button in the "Cells" group on the "Home" tab in the Ribbon. Then click the "Delete Sheet" choice in the button's drop-down menu to delete the selected worksheets.

The following are the instructions for inserting and deleting worksheets in Excel

- To insert a single new worksheet at the end of the current listing of worksheets, click the "New Sheet" button at the right end of the spreadsheet name tabs.
- Alternatively, click the "Home" tab in the Ribbon.
- Then click the "Insert" drop-down button within the "Cells" button group.
- Then select the "Insert Sheet" command from the drop-down menu.
- Alternatively, you could also press the keyboard shortcut of "Shift"+"F11" on your keyboard to automatically insert a new worksheet.
- If using these last two methods, however, note that the new worksheets will always be inserted to the left of the worksheet name that appears in bold when the new sheet is inserted.
- To delete worksheets in Excel, select the worksheet name tabs of the sheet or the sheets you want to delete.

- Click the “Delete” button in the “Cells” group on the “Home” tab in the Ribbon and choose the “Delete Sheet” choice from the button’s drop-down menu.

c) saving worksheet in Excel

Save As

To save your work often or you risk losing your Excel data if the computer crashes or the power goes out. When you save a workbook for the first time, you see the Save As dialog box. The appearance and functionality of the Save As dialog box varies depending on whether you are running Excel 2007, 2010 and so on. Follow these steps to save your workbook when running Excel 2007.

1. Click the Office button and choose Save. The Save As dialog box appears.
2. Click the Browse Folders button (with the triangle pointing downward) to expand the Save As dialog box if necessary.
3. When you expand the Save As dialog box, the dialog box displays the Navigation Pane, where you can select folders listed in the Favorite Links or Folders sections.
4. In the Navigation Pane, select the folder in which you want to save the file. Type a descriptive name for the file in the File Name box.
5. Click the Save button to save the file in the selected folder.

   Save : Follow these steps to save your workbook when running Excel 2007 on Windows XP:

6. Click the Save In drop-down list button and select the folder in which you want to save the file.
7. Type a descriptive name for the file in the File name box.
8. Click the Save button to save the file in the selected folder.

5.4 Page set up and printing worksheet

Before printing a spreadsheet or pivot table, it is used to previewing and adjusting it for better readability. This can be done in different methods. The following are the two methods to page set up and print worksheet in MS Excel 2007.

Page Setup

To modify the page setup

(1) Choose **Page Setup** from the **File** menu. Page Setup allows you to set page options like Margins, Paper Settings, Page Layout and Document Grid. Options for the same are provided through four tabbed sub-menus.

1. **Page :**You can set your page settings here.
2. **Margins:** You can set the margins here.
3. **Header/Footer :**You can define the header and footer of the page here.
4. **Sheet :**You can define your sheet formatting here.

(A) Page Settings

(a) Click here to define the paper orientation;

(b) Click here to define the printer settings;

(c) Click here to see the print preview;

(d) Click here to define the print options;

(e) Click here to define the page scaling;

(f) Click here to define the paper size here;

1. Click here to define the print quality; and

- Click here to define the first page number.

(B) Margin Settings

- Click here to define the **Top** margin setting;
- Click here to define the **Header** margin setting;

(c) Click here to define the **Left** margin setting;

(d) Click here to define the **Right** margin setting;

(e) Click here to define the **Bottom** margin setting;

(f) Click here to define the **Footer** margin setting; and

(g) Click here to define the **Center alignment** of the page.

(C) Header/Footer

(a) Click here to customize your header; and

(b) Click here to customize your footer.

(D) Sheet

(a) Click here to define the print area;

(b) Click here to define thecolumn and rows title;

(c) Click here to print the gridlines, draft quality, row and column heading, cell errors or to print the worksheet in black and white; and

(d) Click here to define the page order in.

Print

To print the document or the template.

1. Choose **Print** from the **File** menu to print your worksheet or a selection of your worksheet.
2. Select the Printer name here.
   1. Click here to change the printing options for the selected printer.
   2. Click here to find the network printer based on the printer name, model etc.
   3. Click here to print your document in a file instead of a printer.
3. Click here to select the page or the page range you want to print.

   (D) Click here to define the number of copies you want to print.
4. Choose the worksheet or the entire workbook that you want to print.

(E) Click here to see preview of the workbook or the worksheet you have selected to print.

5. Click on **OK** Once you have defined your print settings.

Print Preview

Displays the full page as it will be displayed on the paper.

(1) Choose **Print Preview** from the **File** menu to see, how your worksheet or a selection of your worksheet will look like on paper.

(A) This will, howyourworksheet or a selection of your worksheet, will look like on paper.

5.5 Formatting

Excel makes available numerous formatting options to give your worksheet a polished look. You can change the size, colour and angle of fonts, add colour to the borders and backgrounds of cells, and have the format of a cell change, based on its value. You will see that some of the formatting features in MS Excel are same as you have used in MS Word.

USING FORMATTING TOOLBAR

Formatting helps to make our work more presentable. It also helps the viewer/reader to understand the worksheet more easily with respect to its purpose. There are three locations where the Excel 2007 formatting tools are available.

1. In the home tab
2. In the mini toolbar that appears when you right click a range or a cell
3. In the format cells dialog box.

Home Tab and the Formatting Tools: Home tab provides the quick access to the most commonly used options with respect to the formatting requirements. You can select a cell or a range and then use the tool as per the need like Font, Alignment or Number groups. To understand the proper and best use of such tools, only way is to use them and experience their effect. Some of the controls are further expandable like drop-down lists etc. If you see some small arrow along with a button. The button will expand to display the available choices.

In the Mini Toolbar: shortcut menu appears when you right click a cell or a range selection. A mini toolbar also appears above the shortcut menu. The Mini toolbar contains controls for common formatting:

- Font type
- Font Size
- Decrease Font
- Increase Font
- Accounting Number Format
- Comma Style
- Font Color
- Format Painter
- Bold
- Italic
- Center
- Percent Style
- Borders
- Merge And Center
- Increase Decimal
- Decrease Decimal
- Fill Color

Using the Format Cells dialog box:

Although most of the formatting related requirements gets fulfilled by the controls available on the Home tab of the Ribbon, some special types of formatting are fulfilled by using Format cells dialog box.

This dialog box allows to apply more or less any type of formatting style and number formatting. The formats selected from Format Cells Dialog box will be effective to the cells which are selected at the time.

To use Format Cells dialog box, select the cell or a range to apply formatting. Now choose any of the following methods

- Press the combination of Ctrl+1, i.e., Control key and numeric 1 key.
- Click the dialog box launcher in Home → Font, Home → Alignment, or Home → Number. A small down word pointing arrow icon will be displayed to the right. In the Excel Ribbon, these arrow marks are known as dialog box launcher.
  Choose Format Cells from the shortcut menu after you Right-click the selected cell or range.
- Click the More command in some of the drop-down controls in the Ribbon. For example, the Home → Number→ General → More Number Formats… drop-down includes an item named More Number Formats.

5.5.1 Cell FORMATTING

There are six tabs in Format Cells dialog box: Number, Alignment, Font, Border, Patterns and Protection.

Number tab: The data type can be selected from the options on this tab. Select the range you would like to format and click on one of the category number. And apply that style to your numbers.

Alignment tab: These options allow you to change the position and alignment of the data with the cell. The Format Cells dialog box offers you more options than the alignment buttons on the Formatting toolbar. For example, you can change the orientation of the text.

Font tab: All of the font attributes are displayed in this tab including font face, size, style, and effects. Using Formatting toolbar you can bold, italicize, and underline your cell entries. For even more formatting options you can use the Format Cells dialog box.

Border and Pattern tabs:

You can use the Formatting toolbar for adding borders, cell shading, and font colour. These buttons are actually tear-off palettes. When you click on the picture portion of the button, the format of the picture displayed will be applied to the contents of the cell(s) you have selected in the worksheet. You can change the picture displayed on the button by clicking on the button's small drop-down arrow to access the palette of samples from which to choose.

Follow these steps to apply a border and colour to a selection using the options in the Format Cells dialog box.

1. Select Format→Cells to display the Format Cells dialog box.
2. Select the Border tab.
3. In the Presets area, choose None, Outline, or Inside to specify the location for the border.
4. Choose any of the following options for the border:

    - In the Border area, click on any of the buttons to toggle its border.
    - Choose the border's line style in the Style area.
    - If necessary, select a colour for the border in the Color Palette.

5. Select the Patterns tab, and then choose any of the following options:

    - Select a colour for the background of the selection in the Color palette.
    - If necessary, select a pattern for the background of the selection in the Pattern palette.

6. Choose OK to apply the border and colour.

Dates and Times

If you enter the date “January 1, 2001” into a cell on the worksheet, Excel will automatically recognize the text as a date and change the format to “1-Jan-01”. To change the date format, select the Number tab from the Format Cells window. Select “Date” from the Category box and choose the format for the date from the Type box. If the field is a time, select “Time” from the Category box and select the type in the right box. Date and time combinations are also listed. Press OK when finished.

Formatting Columns and Rows:

Sometimes the contents of the columns do not fit in its width. So, to adjust them within the width of the column, we can use AutoFit feature.

Steps to AutoFit Columns:

- Click on Column Header/or any other cell in the column to change its width.
- Click Home tab.
- Select format from the cells group on the ribbon.
- Click on AutoFit Column Width.

Similarly you can apply AutoFit for row also.

- Click on Row Header/or any other cell in the Row to change its Height.
- Click Home tab.
- Select format from the cells group on the ribbon.

Another way of automatically adjusting columns and rows is by way of best fit. Follow the below steps:

- Place your pointer on or near the right edge of a column header of the column you wish to adjust. Notice that in this area your pointer changes to a double-headed arrow.
- Double click your pointer, and the column to the left of it will automatically adjust to fit the data entries within it. Similarly, pointing

to a row header changes pointer to a double headed arrow. Double clicking results in a best fit (taller or shorter rows).

Hide Selected Column(s) or Row(s)

Hide - Hides the selected column(s) or row(s), though the data is still there. Hiding column(s) or row(s) will help you to prevent unwanted changes. To hide a column, follow these steps:

- Select the column you want to hide by clicking on the column header.
- Right Click on the Column to hide and click on the Hide option.

Unhide Selected Column(s) or Row(s)

To unhide the column follow these steps:

- Select the visible range of columns that includes the hidden column(s).
- Now Right Click on the selected Columns. Select Unhide from the pop-up menu.

FORMATTING WORKSHEETS USING CELL STYLES AND APPLYING STYLES:

Excel 2007 provides cell styles to quickly format a cell by choosing from predefined styles. Styles help to give a professional look to your worksheets. In Excel, all styles are cell styles. However, a defined style can be applied to an entire worksheet. Cell styles can include any of the formatting that can be applied to a cell using the options available. We can also define our own cell styles.

- Select the cells to apply a style on.
- Choose Home tab. From Styles group, Click on Cell Styles.

A few of the effects are discussed below

- Comma - Adds commas to the number and two digits beyond a decimal point.
- Comma [0] - Comma style that rounds to a whole number.

- Currency - Formats the number as currency with a dollar sign, commas, and two digits beyond the decimal point.
- Currency [0] - Currency style that rounds to a whole number.
- Normal - Reverts any changes to general number format.
- Percent - Changes the number to a percent and adds a percent sign.

Deleting Styles:

- Right click on the style (say if you want to remove Bad Style)
- Choose delete.

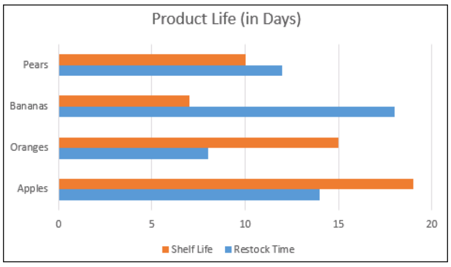
5.6 Creating Charts

In Microsoft Excel, a chart is often called a graph. It is a visual representation of data from a worksheet that can bring more understanding to the data than just looking at the numbers.

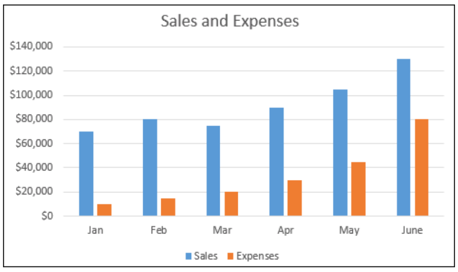
A chart is a powerful tool that allows you to visually display data in a variety of different chart formats such as Bar, Column, Pie, Line, Area, Doughnut, Scatter, Surface, or Radar charts. With Excel, it is easy to create a chart.

The following are some of the types of charts that you can create in Excel.

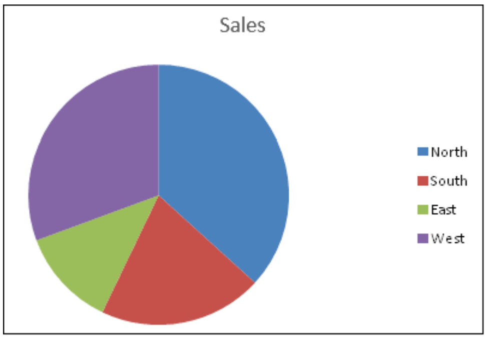
Bar Chart



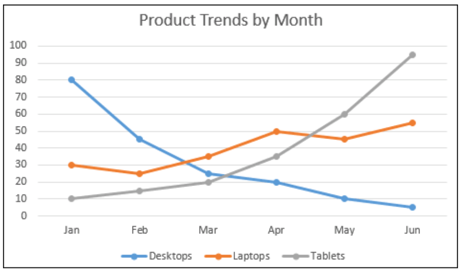
Column Chart



Pie Chart



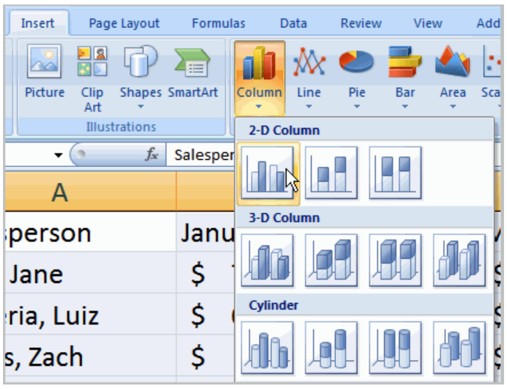
Line Chart



To create a chart follow the steps given below:

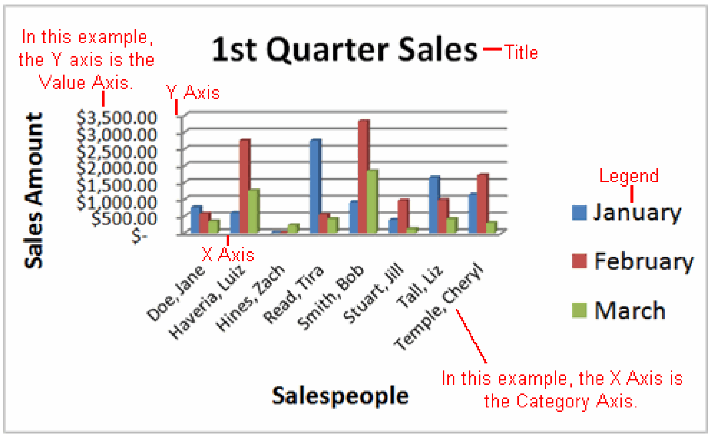
- Select the **worksheet** you want to work with. In this example, we use the **Summary** worksheet.
- Select the **cells** you want to chart, including the **column titles** and **row labels**.
- Click the **Insert** tab.
- Hover over each **Chart option** in the Charts group to learn more about it.

- Select one of the Chart options. In this example, we'll use the Columns command.
- Select a **type of chart** from the list that appears. For this example, we'll use a 2-D Clustered Column. The chart appears in the worksheet.



Identifying the parts of a chart

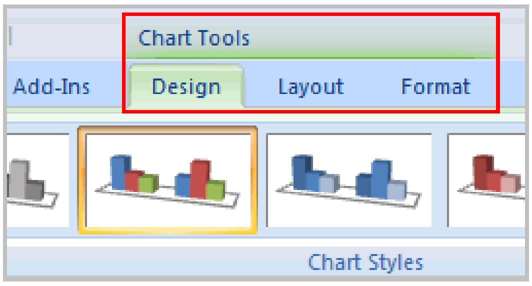
Charts are a **visual representation** of data in a worksheet. Charts make it easy to see comparisons, patterns, and trends in the data.



Source data

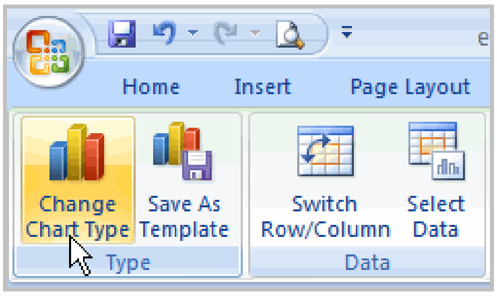
The range of cells that make up a chart. The chart is updated automatically whenever the information in these cells changes.

- **Title:**The title of the chart.
- **Legend:** The chart key, which identifies what each colour on the chart represents.
- **Axis:** The vertical and horizontal parts of a chart. The vertical axis is often referred to as the Y axis, and the horizontal axis is referred to as the X axis.
- **Data series:** The actual charted values, which are usually rows or columns of the source data.
- **Value axis:** The axis that represents the values or units of the source data.
- **Category axis:** The axis identifying each data series.
- **Chart tools:** Once you insert a chart, a new set of **Chart Tools**, arranged into three tabs, will appear above the Ribbon. These are only visible when the chart is selected.



To change the chart type:

- Select the **Design** tab.
- Click the **Change Chart Type** command. A dialog box appears.



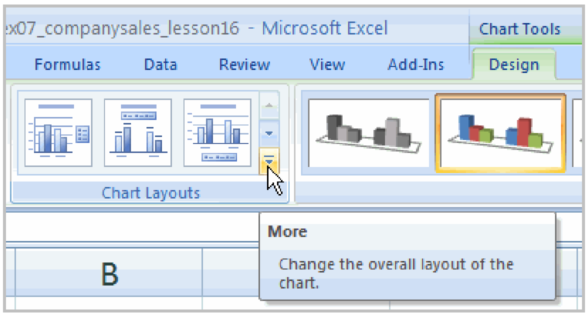
- Select another **chart type**.

- Click OK.

The chart in the example compares each salesperson's monthly sales to his or her other months' sales; however, you can change what is being compared. Just click the **Switch Row/Column Data** command, which will rotate the data displayed on the **x** and **y axes**. To return to the **original view**, click the Switch Row/Column command again.

To change chart layout:

- Select the **Design** tab.
- Locate the **Chart Layouts** group.
- Click the **More** arrow to view all of your layout options.



More

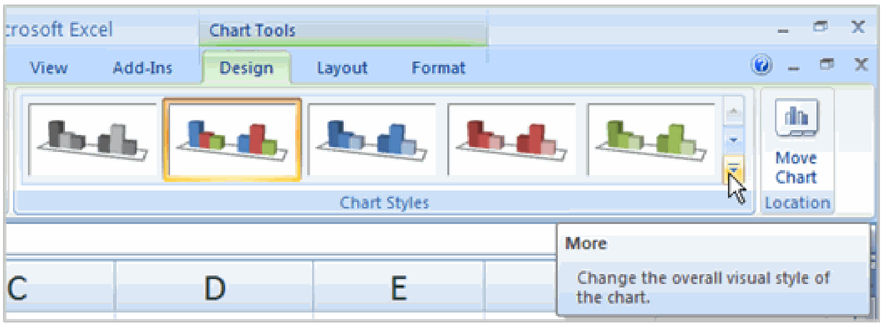
Change the overall layout of the chart.

- Left-click a layout to select it.

If your new layout includes chart titles, axes, or legend labels, just insert your cursor into the text and begin typing to add your own text.

To change chart style:

- Select the **Design** tab.
- Locate the **Chart Style** group.
- Click the **More** arrow to view all of your style options.
- Left-click a style to select it.

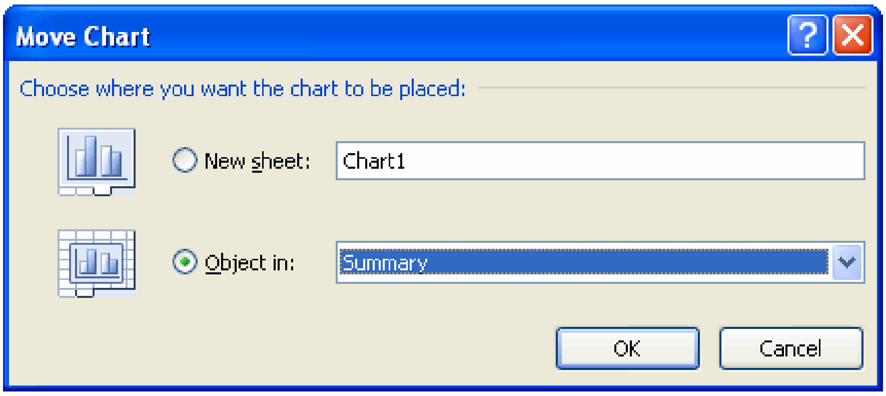


More

Change the overall visual style of the chart.

To move the chart to a different worksheet:

- Select the **Design** tab.
- Click the **Move Chart** command. A dialog box appears. The current location of the chart is selected.
- Select the desired location for the chart (i.e., choose an existing worksheet, or select New Sheet and name it).



Choose where you want the chart to be placed:

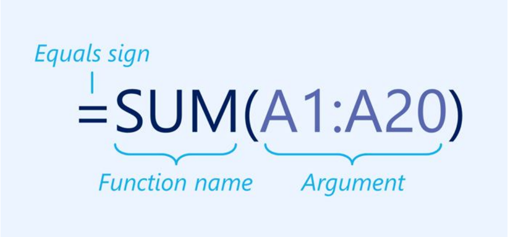
○ New sheet: Chart1

● Object in: Summary

5.7 Worksheets functions

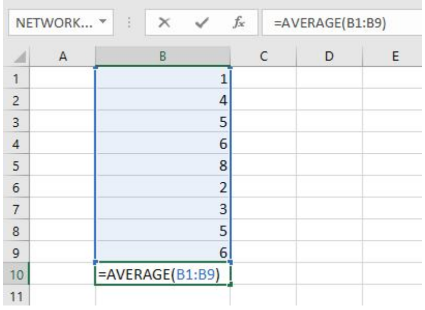
A **function** is a **predefined formula** that performs calculations using specific values in a particular order. Excel includes many common functions that can be used to quickly find the **sum**, **average**, **count**, **maximum value**, and **minimum value** for a range of cells. In order to use functions correctly, you'll need to understand the different **parts of a function** and how to create **arguments** to calculate values and cell references.

5.7.1 Function: In order to work correctly, a function must be written a specific way, which is called the **syntax**. The basic syntax for a function is the **equals sign (=)**, the **function name** (SUM, for example), and one or more **arguments**. Arguments contain the information you want to calculate. The function in the example below would add the values of the cell range A1:A20.



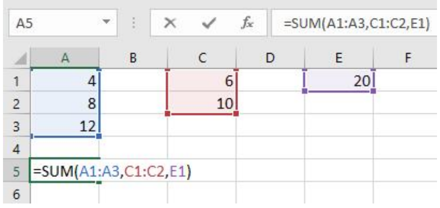
Working with arguments: Arguments can refer to both **individual cells** and **cell ranges** and must be enclosed within **parentheses**. You can include one argument or multiple arguments, depending on the syntax required for the function.

For example, the function **=AVERAGE(B1:B9)** would calculate the **average** of the values in the cell range B1:B9. This function contains only one argument.


NETWORK... =AVERAGE(B1:B9)

	A	B	C	D	E
1		1			
2		4			
3		5			
4		6			
5		8			
6		2			
7		3			
8		5			
9		6			
10		=AVERAGE(B1:B9)			
11					

Multiple arguments must be separated by a **comma**. For example, the function **=SUM(A1:A3, C1:C2, E1)** will **add** the values of all of the cells in the three arguments.


A5 =SUM(A1:A3,C1:C2,E1)

	A	B	C	D	E	F
1	4		6		20	
2	8		10			
3	12					
4						
5	=SUM(A1:A3,C1:C2,E1)					
6						

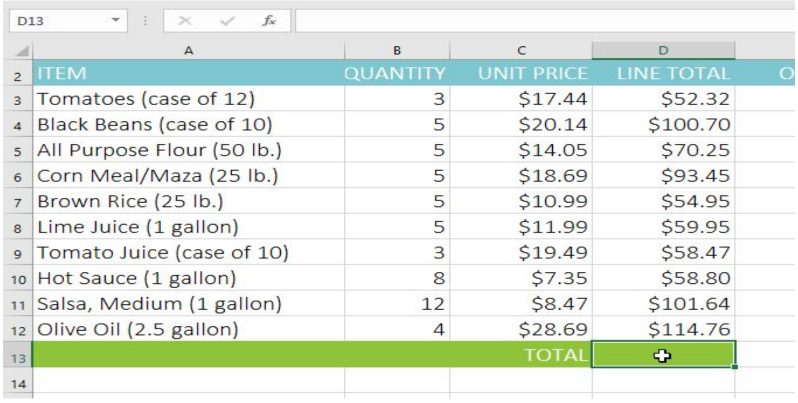
Creating a function: There are a variety of functions available in Excel. Here are some of the most common functions**.**

- **SUM**: This function **adds** all of the values of the cells in the argument.
- **AVERAGE**: This function determines the **average** of the values included in the argument. It calculates the sum of the cells and then divides that value by the number of cells in the argument.
- **COUNT**: This function **counts** the number of cells with numerical data in the argument. This function is useful for quickly counting items in a cell range.
- **MAX**: This function determines the **highest cell value** included in the argument.
- **MIN**: This function determines the **lowest cell value** included in the argument.

To create a function using the AutoSum command:

The **AutoSum** command allows you to automatically insert the most common functions into your formula, including SUM, AVERAGE, COUNT, MIN, and MAX. In the example below, we'll use the **SUM** function to calculate the **total cost** for a list of recently ordered items.

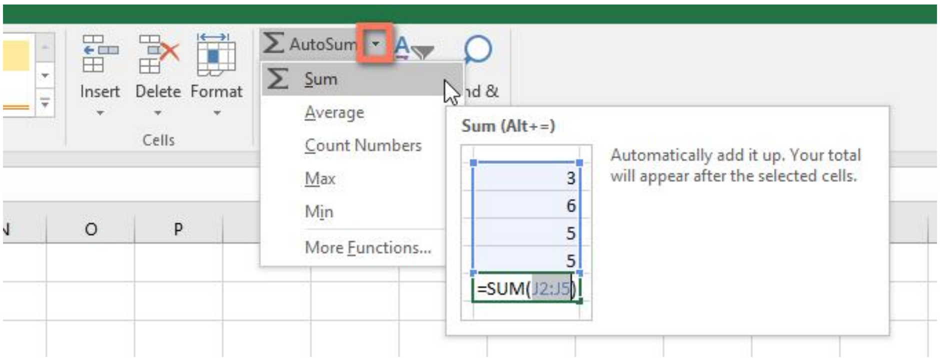
1. Select the **cell** that will contain the function. In our example, we'll select cell **D13**.



D13

	A	B	C	D
2	ITEM	QUANTITY	UNIT PRICE	LINE TOTAL
3	Tomatoes (case of 12)	3	$17.44	$52.32
4	Black Beans (case of 10)	5	$20.14	$100.70
5	All Purpose Flour (50 lb.)	5	$14.05	$70.25
6	Corn Meal/Maza (25 lb.)	5	$18.69	$93.45
7	Brown Rice (25 lb.)	5	$10.99	$54.95
8	Lime Juice (1 gallon)	5	$11.99	$59.95
9	Tomato Juice (case of 10)	3	$19.49	$58.47
10	Hot Sauce (1 gallon)	8	$7.35	$58.80
11	Salsa, Medium (1 gallon)	12	$8.47	$101.64
12	Olive Oil (2.5 gallon)	4	$28.69	$114.76
13			TOTAL	
14				

2. In the **Editing** group on the **Home** tab, click the **arrow** next to the **AutoSum** command. Next, choose the **desired function** from the drop-down menu. In our example, we'll select **Sum**.

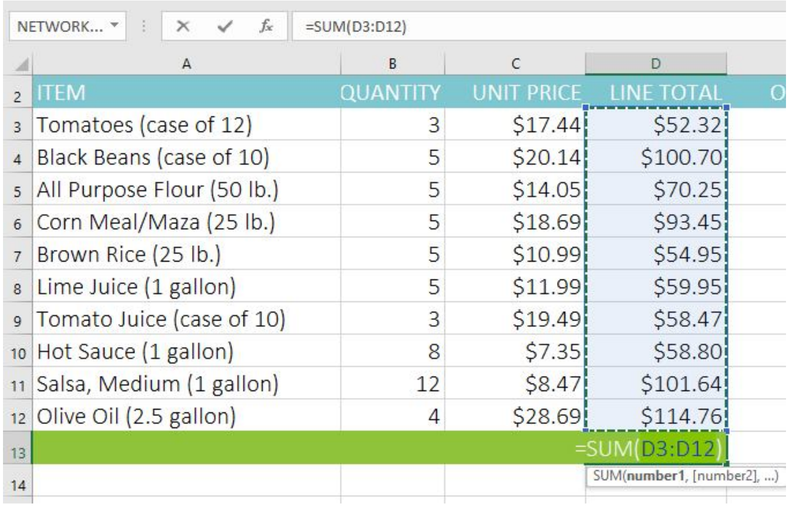


Sum (Alt+=)

Automatically add it up. Your total will appear after the selected cells.

=SUM(J2:J5)

3. Excel will place the **function** in the cell and automatically select a **cell range** for the argument. In our example, cells **D3:D12** were selected automatically; their values will be **added** to calculate the total cost. If Excel selects the wrong cell range, you can manually enter the desired cells into the argument.

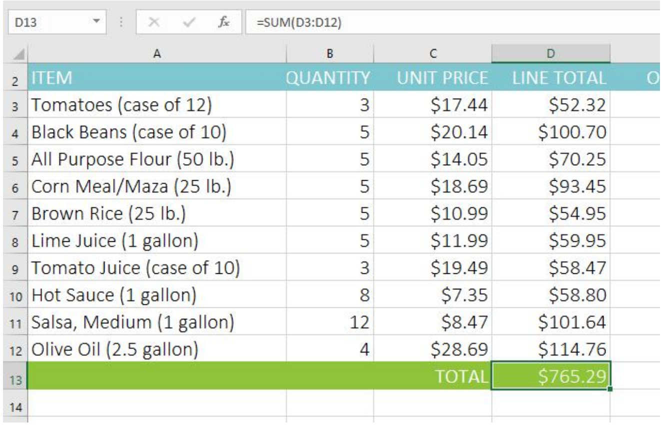


NETWORK... =SUM(D3:D12)

	A	B	C	D
2	ITEM	QUANTITY	UNIT PRICE	LINE TOTAL
3	Tomatoes (case of 12)	3	$17.44	$52.32
4	Black Beans (case of 10)	5	$20.14	$100.70
5	All Purpose Flour (50 lb.)	5	$14.05	$70.25
6	Corn Meal/Maza (25 lb.)	5	$18.69	$93.45
7	Brown Rice (25 lb.)	5	$10.99	$54.95
8	Lime Juice (1 gallon)	5	$11.99	$59.95
9	Tomato Juice (case of 10)	3	$19.49	$58.47
10	Hot Sauce (1 gallon)	8	$7.35	$58.80
11	Salsa, Medium (1 gallon)	12	$8.47	$101.64
12	Olive Oil (2.5 gallon)	4	$28.69	$114.76
13				=SUM(D3:D12)
14				

SUM(**number1**, [number2], ...)

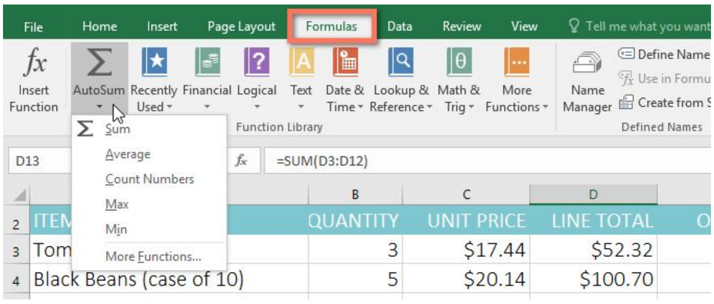
4. Press **Enter** on your keyboard. The function will be **calculated**, and the **result** will appear in the cell. In our example, the sum of D3:D12 is **$765.29**.



D13 =SUM(D3:D12)

	A	B	C	D
2	ITEM	QUANTITY	UNIT PRICE	LINE TOTAL
3	Tomatoes (case of 12)	3	$17.44	$52.32
4	Black Beans (case of 10)	5	$20.14	$100.70
5	All Purpose Flour (50 lb.)	5	$14.05	$70.25
6	Corn Meal/Maza (25 lb.)	5	$18.69	$93.45
7	Brown Rice (25 lb.)	5	$10.99	$54.95
8	Lime Juice (1 gallon)	5	$11.99	$59.95
9	Tomato Juice (case of 10)	3	$19.49	$58.47
10	Hot Sauce (1 gallon)	8	$7.35	$58.80
11	Salsa, Medium (1 gallon)	12	$8.47	$101.64
12	Olive Oil (2.5 gallon)	4	$28.69	$114.76
13			TOTAL	$765.29
14				

The **AutoSum** command can also be accessed from the **Formulas** tab on the **Ribbon**.



D13 =SUM(D3:D12)

	A	B	C	D
2	ITEM	QUANTITY	UNIT PRICE	LINE TOTAL
3	Tom	3	$17.44	$52.32
4	Black Beans (case of 10)	5	$20.14	$100.70

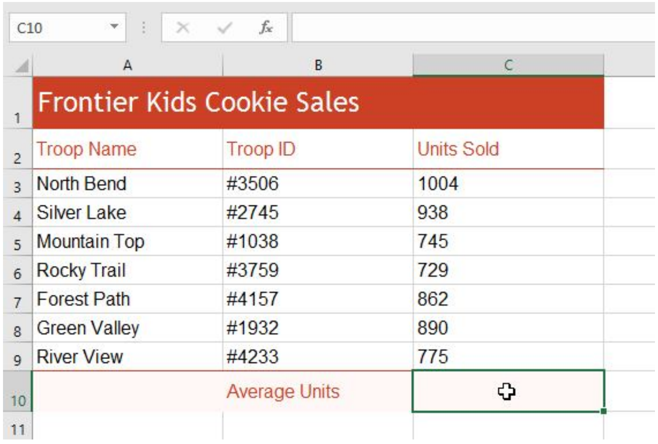
You can also use the **Alt+=** keyboard shortcut instead of the AutoSum command. To use this shortcut, hold down the **Alt** key and then press the **equals sign**.

To enter a function manually:

If you already know the function name, you can easily type it yourself. In the example below (a tally of cookie sales), we'll use the **AVERAGE** function to calculate the **average number of units sold** by each troop.

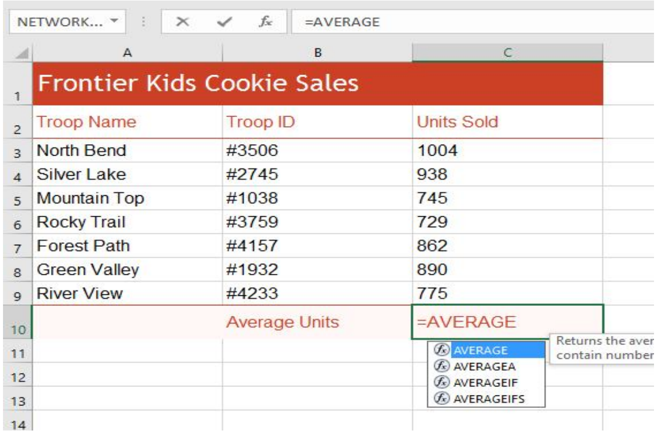
1. Select the **cell** that will contain the function. In our example, we'll select cell **C10**.

2.



	A	B	C
1	Frontier Kids Cookie Sales		
2	Troop Name	Troop ID	Units Sold
3	North Bend	#3506	1004
4	Silver Lake	#2745	938
5	Mountain Top	#1038	745
6	Rocky Trail	#3759	729
7	Forest Path	#4157	862
8	Green Valley	#1932	890
9	River View	#4233	775
10		Average Units	
11			

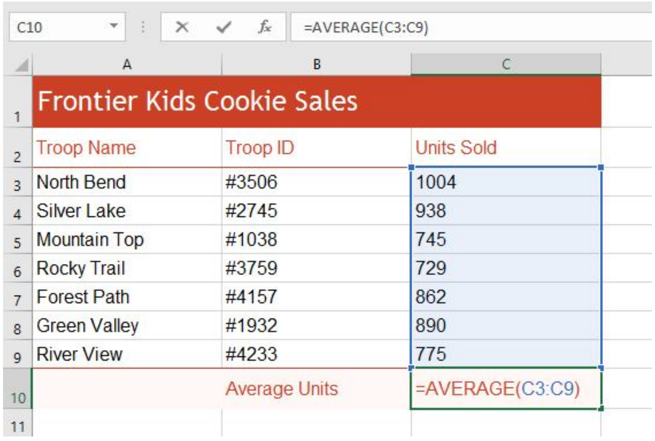
3. Type the **equals sign (=),** and enter the desired **function name**. You can also select the desired function from the list of **suggested functions** that appears below the cell as you type. In our example, we'll type **=AVERAGE**.



NETWORK... =AVERAGE

	A	B	C
1	Frontier Kids Cookie Sales		
2	Troop Name	Troop ID	Units Sold
3	North Bend	#3506	1004
4	Silver Lake	#2745	938
5	Mountain Top	#1038	745
6	Rocky Trail	#3759	729
7	Forest Path	#4157	862
8	Green Valley	#1932	890
9	River View	#4233	775
10		Average Units	=AVERAGE
11			
12			
13			
14			

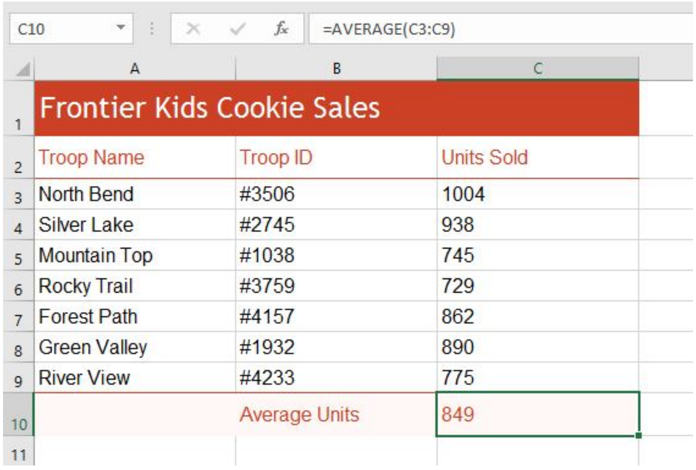
4. Enter the **cell range** for the argument inside **parentheses**. In our example, we'll type **(C3:C9)**. This formula will add the values of cells C3:C9, then divide that value by the total number of values in the range.



C10 =AVERAGE(C3:C9)

	A	B	C
1	Frontier Kids Cookie Sales		
2	Troop Name	Troop ID	Units Sold
3	North Bend	#3506	1004
4	Silver Lake	#2745	938
5	Mountain Top	#1038	745
6	Rocky Trail	#3759	729
7	Forest Path	#4157	862
8	Green Valley	#1932	890
9	River View	#4233	775
10		Average Units	=AVERAGE(C3:C9)
11			

5. Press **Enter** on your keyboard. The function will be calculated, and the **result** will appear in the cell. In our example, the average number of units sold by each troop is **849**.



C10 =AVERAGE(C3:C9)

	A	B	C
1	Frontier Kids Cookie Sales		
2	Troop Name	Troop ID	Units Sold
3	North Bend	#3506	1004
4	Silver Lake	#2745	938
5	Mountain Top	#1038	745
6	Rocky Trail	#3759	729
7	Forest Path	#4157	862
8	Green Valley	#1932	890
9	River View	#4233	775
10		Average Units	849
11			

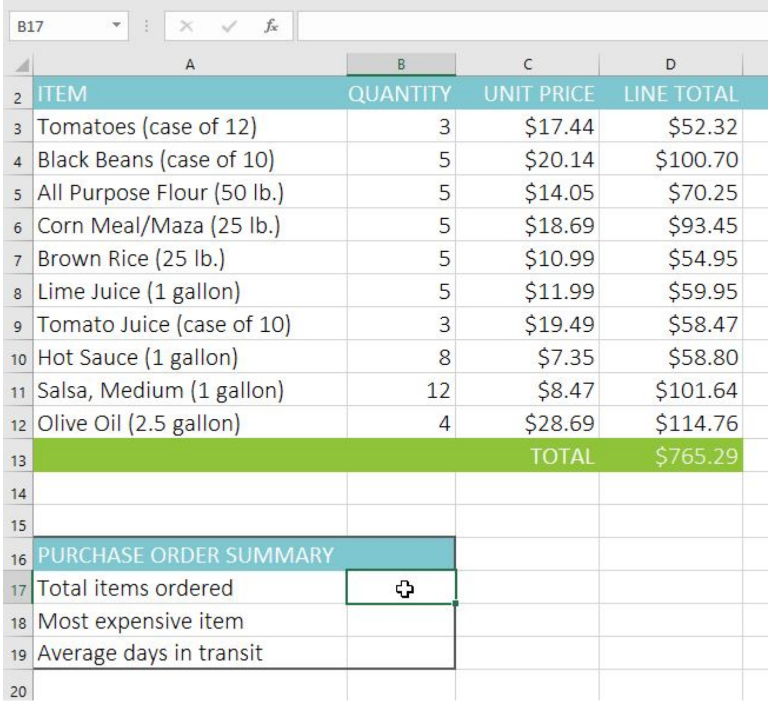
The Function Library

While there are hundreds of functions in Excel, the ones you'll use the most will depend on the **type of data** your workbooks contain. There's no need to learn every single function, but exploring some of the different **types** of functions will help you as you create new projects. You can even use the **Function Library** on the **Formulas** tab to browse functions by category, such as **Financial**, **Logical**, **Text**, and **Date & Time**. To access the **Function Library**, select the **Formulas** tab on the **Ribbon**.

To insert a function from the Function Library:

In the example below, we'll use the COUNTA function to count the total number of items in the **Items** column. Unlike COUNT, **COUNTA** can be used to tally cells that contain data of any kind, not just numerical data.

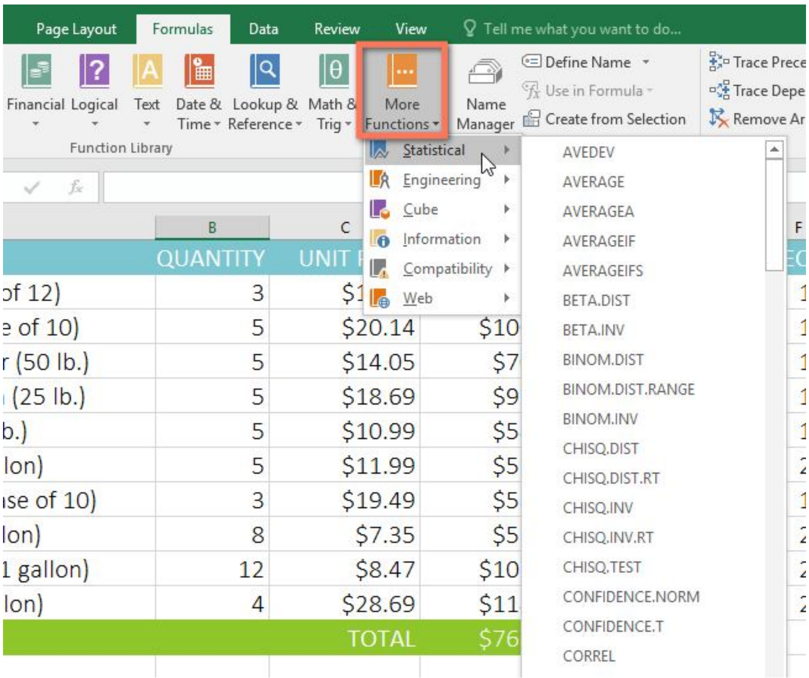
1. Select the **cell** that will contain the function. In our example, we'll select cell **B17**.



	A	B	C	D
2	ITEM	QUANTITY	UNIT PRICE	LINE TOTAL
3	Tomatoes (case of 12)	3	$17.44	$52.32
4	Black Beans (case of 10)	5	$20.14	$100.70
5	All Purpose Flour (50 lb.)	5	$14.05	$70.25
6	Corn Meal/Maza (25 lb.)	5	$18.69	$93.45
7	Brown Rice (25 lb.)	5	$10.99	$54.95
8	Lime Juice (1 gallon)	5	$11.99	$59.95
9	Tomato Juice (case of 10)	3	$19.49	$58.47
10	Hot Sauce (1 gallon)	8	$7.35	$58.80
11	Salsa, Medium (1 gallon)	12	$8.47	$101.64
12	Olive Oil (2.5 gallon)	4	$28.69	$114.76
13			TOTAL	$765.29
14				
15				
16	PURCHASE ORDER SUMMARY			
17	Total items ordered			
18	Most expensive item			
19	Average days in transit			
20				

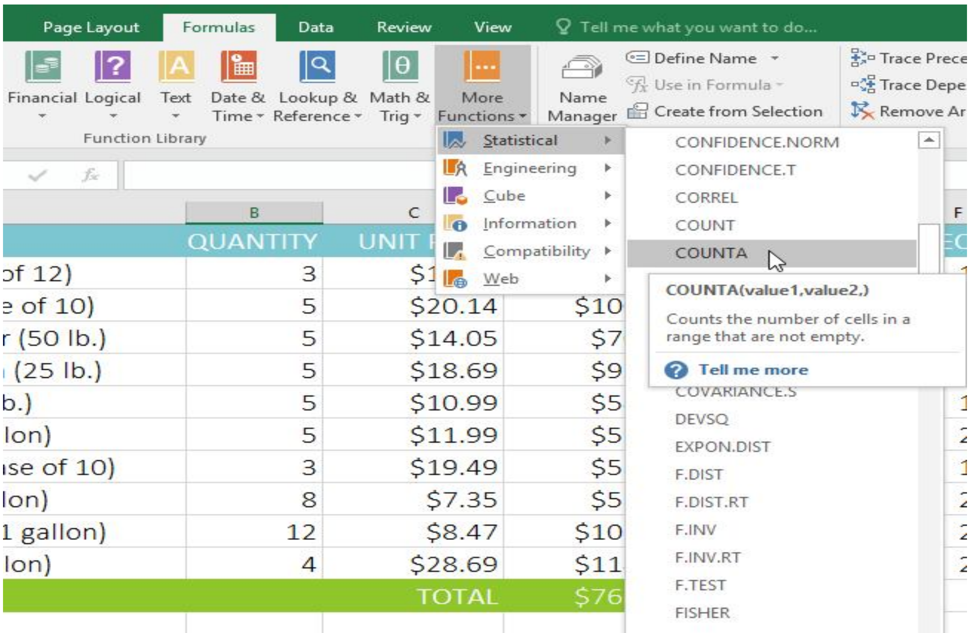
2. Click the **Formulas** tab on the **Ribbon** to access the **Function Library**.

3. From the **Function Library** group, select the desired **function category**. In our example, we'll choose **More Functions**, then hover the mouse over **Statistical**.



	B	C	
	QUANTITY	UNIT F	
of 12)	3	$1	
e of 10)	5	$20.14	$10
r (50 lb.)	5	$14.05	$7
(25 lb.)	5	$18.69	$9
b.)	5	$10.99	$5
lon)	5	$11.99	$5
ise of 10)	3	$19.49	$5
lon)	8	$7.35	$5
1 gallon)	12	$8.47	$10
lon)	4	$28.69	$11
		TOTAL	$76

4. Select the **desired function** from the drop-down menu. In our example, we'll select the **COUNTA** function, which will count the number of cells in the **Items** column that are not empty.



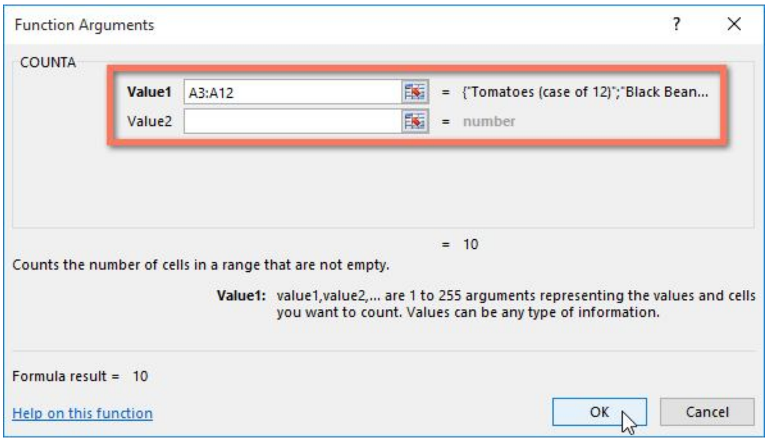
COUNTA(value1,value2,)

Counts the number of cells in a range that are not empty.

	B	C	
	QUANTITY	UNIT I	
of 12)	3	$1	
e of 10)	5	$20.14	$10
r (50 lb.)	5	$14.05	$7
(25 lb.)	5	$18.69	$9
b.)	5	$10.99	$5
lon)	5	$11.99	$5
ıse of 10)	3	$19.49	$5
lon)	8	$7.35	$5
1 gallon)	12	$8.47	$10
lon)	4	$28.69	$11
		TOTAL	$76

5. The **Function Arguments** dialog box will appear. Select the **Value1** field, then enter or select the desired cells. In our example, we'll enter the cell range **A3:A12**. You may continue to add arguments in the **Value2** field, but in this case we only want to count the number of cells in the cell range **A3:A12**.

6. When you're satisfied, click **OK**.



Function Arguments

COUNTA

Value1 A3:A12 = {"Tomatoes (case of 12)";"Black Bean...

Value2 = number

= 10

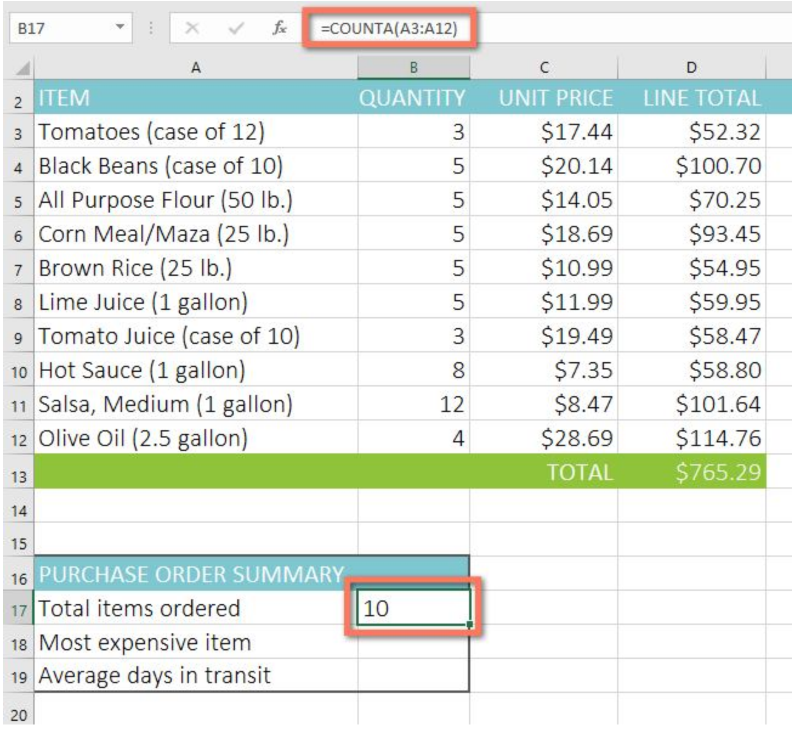
Counts the number of cells in a range that are not empty.

Value1: value1,value2,... are 1 to 255 arguments representing the values and cells you want to count. Values can be any type of information.

Formula result = 10

Help on this function

7. The function will be **calculated**, and the **result** will appear in the cell. In our example, the result shows that a total of **10 items** were ordered.



B17 =COUNTA(A3:A12)

	A	B	C	D
2	ITEM	QUANTITY	UNIT PRICE	LINE TOTAL
3	Tomatoes (case of 12)	3	$17.44	$52.32
4	Black Beans (case of 10)	5	$20.14	$100.70
5	All Purpose Flour (50 lb.)	5	$14.05	$70.25
6	Corn Meal/Maza (25 lb.)	5	$18.69	$93.45
7	Brown Rice (25 lb.)	5	$10.99	$54.95
8	Lime Juice (1 gallon)	5	$11.99	$59.95
9	Tomato Juice (case of 10)	3	$19.49	$58.47
10	Hot Sauce (1 gallon)	8	$7.35	$58.80
11	Salsa, Medium (1 gallon)	12	$8.47	$101.64
12	Olive Oil (2.5 gallon)	4	$28.69	$114.76
13			TOTAL	$765.29
14				
15				
16	PURCHASE ORDER SUMMARY			
17	Total items ordered	10		
18	Most expensive item			
19	Average days in transit			
20				

5.8 Summary

- MS Excel is a powerful spreadsheet program through which you can create tables of any kind.
- Excel features: You can format cells/data according to condition and can get glimpse of entire data within seconds. Pivot tables, lookup formulas, power pivot, add header and footer, find and replace, pass word protection, data filtering, built in formulae and many more.
- **Formatting Columns and Rows:**
- Sometimes the contents of the columns do not fit in its width. So, to adjust them within the width of the column, we can use AutoFit feature.
- **Steps to AutoFit Columns**:
  - Click on Column Header/or any other cell in the column to change its width.
  - Click Home tab.
  - Select format from the cells group on the ribbon.

    - Click on AutoFit Column Width.

- In Microsoft Excel, a chart is often called a graph. It is a visual representation of data from a worksheet that can bring more understanding to the data than just looking at the numbers.

5.9. SAQ

1. What is Excel? List features of Excel?
2. Explain how to create, insert and save a worksheet?
3. Create a work sheet for student, where student annual marks are calculated and prepare a pi graph for the student details?
4. What is formatting a cell? Explain?
5. Write a short note on functions in Excel?

UNIT – VI Introduction to C language

Structure

6.0 Objective

The main objective of this unit is to make clear about how C programming Language came in to existence, what are variable, constants? How to declare and used? what are different types of operators and expression used in C language.

6.1Introduction

This chapter gives an introduction to basic C language concepts. These concepts include the C character set (alphabet), quantitative (Variables and constants) and qualitative (Data Types) concepts of C words.

C is a computer programming Language. Programming Languages are used to solve problems on computers using programs. A program is a series of instructions, which is used to solve a specific problem. These programs are tested for known inputs. If they work fine, they are used for new and complex data for which answers are not known.

The advantages of solving problems using Computers programs are given below.

- Automation (No human intervention)
- Fastness
- Accuracy

Before attempting to solve a problem on computer, we must understand that a computer can neither think nor can make judgments on its own. No computer can analyze the problem and identify its inputs to process so as to derive the output. But it could execute a given set of instructions. The instructions are of different types which are listed below.

(i) Input statements – which get input from input devices.

(ii) Output statements – which print contents of memory locations to some output devices.

(iii) Arithmetic statements – which evaluate arithmetic calculations and store the results appropriately in the prescribed memory locations.

(iv) Logical operations – which compare two quantities in terms of relative ness.

(v) Conditional statements – execute statements depending on the result of evaluated logical operations embedded in the conditional statement.

6.2 Overview of C Language

C is a structured programming language developed by Dennis Ritchie in 1973 at Bell Laboratories. It is one of the most popular computer languages today because of its structure, high-level abstraction, machine independent feature.

C language was developed with UNIX operating system, so it is strongly associated with UNIX, which is one of the most popular network operating system in use today and heart of internet data superhighway.C has now become a widely used professional language for various reasons:

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

6.2.1 History of C

C language has evolved from three different structured language ALGOL, BCPL and B Language. It uses many concepts from these languages and introduced many new concepts such as data types, struct, pointer. In 1988, the language was formalized by **American National Standard Institute** (ANSI). In 1990, a version of C language was approved by the **International Standard Organization** (ISO) and that version of C is also referred to as C89.

6.2.2 Importance of C

C was initially used for system development work, particularly the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as the code written in assembly language. Some examples of the use of C might be:

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities

6.2.3 Basic Structure of C programs and executing a C program

The basic structure of the C programming language is studied in this part of the lesson. A C program basically consists of the following parts:

- Preprocessor Commands
- Functions
- Variables
- Statements & Expressions
- Comments

Let us consider a simple C program that would print the words "Hello World":

```
#include <stdio.h>

main()
{
   /* my first program in C */
printf(" Hello world ! \n");
    return 0;
}
```

Now let us consider various steps that are involved in the above simple C program:

1. The first line of the program #include is a preprocessor command, which tells a C compiler to include stdio.h file before going to actual compilation.
2. The next line int main() is the main function where the program execution begins.
3. The next line /*...*/ will be ignored by the compiler and it has been put to add additional comments in the program. So such lines are called comments in the program.
4. The next line printf(...) is another function available in C which causes the message "Hello World!" to be displayed on the screen.

5. The next line return 0; terminates the main() function and returns the value 0.

Compile and execution of C program

Consider the following steps how to save the source code in a file, and how to compile and run it.

- Open a text editor and add the write the above code.
- Save the file as hello.c by clicking F2
- Open a command prompt and go to the directory where you have saved the file.
- Compile the program using Alt + F9 for compile
- Press Ctrl + F9 to Run or select Run > Run in menu bar
- Alt +F5 to view the output of the program at the output screen.

Out Put

```
Hello world!
```

Semicolons: The semicolon in C language acts as terminator i.e., each individual statement in C language must be ended with a semicolon. It indicates the end of the logical statement.

Comments: Comments are like helping text in your C program and they are ignored by the compiler. They start with /* and terminate with the characters */ as shown in the above example.

6.3 Constants, Variables and Data types

Character denotes any alphabet, digit or special symbol used to represent information. These characters can be combined to form variables. C uses constants, variables, operators, keywords and expressions as building blocks to form a basic C program.

6.3.1 Character set:

The character set is the fundamental raw material of any language and they are used to represent information. Like natural languages, computer language will also have well defined characterset, which is useful to build the programs. The characters in C are grouped into the following **two** categories:

1. **Source character set**
    a. Alphabets
    b. Digits
    c. Special Characters
    d. White Spaces

2. **Execution character set**
    a. Escape Sequence

Source character set

ALPHABETS :	Uppercase letters	A-Z
	Lowercase letters	a-z

DIGITS : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

SPECIAL CHARACTERS

~	tilde	%	percent sign
|	vertical bar	@	at symbol
+	plus sign	<	less than
_	underscore	-	minus sign
>	greater than	^	caret
#	number sign	=	equal to
&	ampersand	$	dollar s ign
/	slash	(	left parenthesis
*	asterisk	\	back slash
)	right parenthesis	′	apostrophe
:	colon	[	left bracket
"	quotation mark	;	semicolon

! exclamation mark	, comma
] right bracket	. dot operator
{ left flower brace	? Question mark
} right flower brace	

6.3.2 C Tokens, Key words and Identifiers

C tokens, identifiers and keywords are the basics in C program.

1. **C Tokens:**

   C tokens are the basic building blocks in a C language which are constructed together to write a C program. Each and every smallest individual units in C program are know as C tokens. The following are the six types of tokens in C.

Key words	(eg: int, while)
Identifiers	(eg: main, total)
Constants	(eg: 10,20)
Strings	(eg:"total", "hello")
Special symbols	(eg: {},())
Operators	(eg:+, - , * , /)

The following is an example for C tokens

```
int main()
{
  int x, y, total;
  x = 10, y = 20;
  total = x + y;
printf ("Total = %d \n", total);
}
```

Where

- main – identifier

- {,},(,) – delimiter
- Int - keyword
- x,y, total – identifier
- main, {,}, (,), int, x, y , total - tokens

2. **Keywords:**

Keywords are predefined; reserved words used in programming that have special meanings to the compiler. Keywords are part of the syntax and they cannot be used as an identifier. As C is a case sensitive language, all keywords must be written in lowercase. Here is a list of all keywords allowed in ANSI - C.

Keywords in C Language			
auto	double	int	Struct
break	else	long	Switch
case	enum	register	Typedef
char	extern	return	Union
continue	for	signed	Void
do	if	static	While
default	goto	sizeof	Volatile
const	float	short	Unsigned

3. **Identifiers:**

Identifiers are names for entities in a C program, such as variables, arrays, functions, structures, unions and labels. An identifier can be composed only of uppercase, lowercase letters, underscore and digits, but should start only with an alphabet or an underscore. If the identifier is not used in an external link process, then it is called as internal. **Example:** Local variable. If the identifier is used in an external link process, then it is called as external. **Example:** Global variable.

An identifier is a string of alphanumeric characters that begins with an alphabetic character or an underscore character that are used to represent various programming elements such as variables, functions, arrays, structures, unions and so on. Actually, an identifier is a user-defined word. There are 53 characters, to represent identifiers. They are 52 alphabetic characters (i.e., both uppercase and lowercase alphabets) and the underscore character. The underscore character is considered as a letter in identifiers. The underscore character is usually used in the middle of an identifier. There are 63 alphanumeric characters, i.e., 53 alphabetic characters and 10 digits (i.e., 0-9).

Rules for constructing identifiers:

1. The first character in an identifier must be an alphabet or an underscore and can be followed only by any number alphabets, or digits or underscores.
2. They must not begin with a digit.
3. Uppercase and lowercase letters are distinct. That is, identifiers are case sensitive.
4. Commas or blank spaces are not allowed within an identifier.
5. Keywords cannot be used as an identifier.
6. Identifiers should not be of length more than 31 characters.
7. Identifiers must be meaningful, short, quickly and easily typed and easily read.

Valid identifiers: total sum average _x y_ mark_1 x1

Invalid identifiers: 1x - should begins with a digit
char - reserved word
x+y - special character

Note: Underscore character is usually used as a link between two words in long identifiers.

kinds of identifiers

C defines two kinds of identifiers:

- **Internal**
- **External**

Internal identifier: If the identifier is used in an external link process, then it is called as **external**. These identifiers are also known as **external names**; include function names and global variable names that are shared between source files. It has at least 63 significant characters.

External identifier: If the identifier is not used in an external link process, then it is called as **internal**. These identifiers are also known as **internal names**; includes the names of local variables. It has at least 31 significant characters.

SQA

1. Write short notes on: Identifiers.

2. What are identifiers in C? Give examples?

3. Define identifier. What are the rules to be followed for identifiers?

4. What are the different types of identifiers in C?

5. What is an internal identifier?

6. What is an external identifier?

6.3.3 Constants:

Constants refer to fixed values that the program may not alter during its execution. These fixed values are also called literals. Constants can be of any of the basic data types like an integer constant, a floating constant, a character constant, or a string literal. There are enumeration constants as well. Constants are treated just like regular variables except that their values cannot be modified after their definition.

Integer Literals:

Integer Literals An integer literal can be a decimal, octal, or hexadecimal constant. A prefix specifies the base or radix: 0x or 0X for hexadecimal, 0 for octal, and nothing for decimal. An integer literal can also have a suffix that is a combination of U and L, for unsigned and long, respectively. The suffix can be uppercase or lowercase and can be in any order. following are the some examples of integer literals:

212 /* Legal */
215u /* Legal */
0xFeeL /* Legal */
078 /* Illegal: 8 is not an octal digit */
032UU /* Illegal: cannot repeat a suffix */

Floating-point literals:

floating-point literal has an integer part, a decimal point, a fractional part, and an exponent part. You can represent floating point literals either in decimal form or exponential form. While representing decimal form, you must include the decimal point, the exponent, or both; and while representing exponential form, you must include the integer part, the fractional part, or both. The signed exponent is introduced by e or E. Some examples of floating-point literals are given below:

3.14159 /* Legal */
314159E-5L /* Legal */
510E /* Illegal: incomplete exponent */
210f /* Illegal: no decimal or exponent */

.e55	/* Illegal: missing integer or fraction */

Character Constants:

Character Constants Character literals are enclosed in single quotes, e.g., 'x' can be stored in a simple variable of char type. A character literal can be a plain character (e.g., 'x'), an escape sequence (e.g., '\t'), or a universal character (e.g., '\u02C0'). There are certain characters in C that represent special meaning when preceded by a backslash, for example, newline (\n) or tab (\t).

Meaning	Escape sequence
\\	\ character
\’ character	\’ ‘ character
\? character	\? Character
\a	Alert or bell
\b	Back space
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal return
\ooo	Vertical tab
\xhh…	Hexadecimal number of one or more digits

Consider small example to show how above character constants work

```
# include <stdio.h>
main()
{
printf("Hello\tWorld\n\n");
```

```
  return 0;
}
```

Out put:

```
Hello World
```

String Literals:

String Literals String literals or constants are enclosed in double quotes "". A string contains characters that are similar to character literals: plain characters, escape sequences, and universal characters.

Defining Constants: There are two simple ways in C to define constants:

- Using #define preprocessor
- Using const keyword

The #define Preprocessor Given below is the form to use #define preprocessor to define a constant. consider the following example

```
#include <stdio.h>

#define LENGTH 10
#define WIDTH  5
#define NEWLINE '\n'

main()
{

  int area;

  area = LENGTH * WIDTH;
printf("value of area : %d", area);
printf("%c", NEWLINE);

  return 0;
}
```

When the above code is compiled and executed, it produces the following output:

```
value of area : 50
```

6.3.4 Variables, declaration of variables and assigning values to variables

A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in C has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable. The name of a variable can be composed of letters, digits, and the underscore character. It must begin with either a letter or an underscore. Upper and lowercase letters are distinct because C is case-sensitive. Based on the basic types explained in the previous chapter, there will be the following basic variable types:

Type	Description
Char	Typically a single octet(one byte). This is an integer type
Int	The most natural size of integer for the machine
Float	A single-precision floating point value
Double	A double-precision floating point value
Void	Represents the absence of type

Defining a variable in C

A variable definition tells the compiler where and how much storage to create for the variable. A variable definition specifies a data type and contains a list of one or more variables of that type as follows:

```
type variable_list;
```

Here, type must be a valid C data type including char, w_char, int, float, double, bool, or any user-defined object; and variable_list may consist of one or more identifier names separated by commas. Some valid declarations are shown here:

```
int    i, j, k;
char   c, ch;
float  f, salary;
double d;
```

The line int i, j, k; declares and defines the variables i, j and k; which instruct the compiler to create variables named i, j, and k of type int. Variables can be initialized (assigned an initial value) in their declaration. The initializer consists of an equal sign followed by a constant expression as follows:

```
type variable_name = value;
```

consider some examples:

```
extern int d = 3, f = 5;    // declaration of d and f.
int d = 3, f = 5;           // definition and initializing d and f.
byte z = 22;                // definition and initializes z.
char x = 'x';               // the variable x has the value 'x'.
```

For definition without an initializer: variables with static storage duration are implicitly initialized with NULL (all bytes have the value 0); the initial value of all other variables are undefined. The following is example how variables are declare and accessed in C language.

```
include<stdio.h>
void test();

int main()
{
  int m = 22, n = 44;
      // m, n are local variables of main function
      /*m and n variables are having scope
      within this main function only.
      These are not visible to test funtion.*/
      /* If you try to access a and b in this function,
      you will get 'a' undeclared and 'b' undeclared error */
  printf("\nvalues : m = %d and n = %d", m, n);
  test();
}

void test()
{
  int a = 50, b = 80;
      // a, b are local variables of test function
      /*a and b variables are having scope
      within this test function only.
      These are not visible to main function.*/
      /* If you try to access m and n in this function,
      you will get 'm' undeclared and 'n' undeclared
      error */
  printf("\nvalues : a = %d and b = %d", a, b);
}
```

Out put:

```
 Values: m=22 and n=44 (for the first program)
 Values: a=50 and b=80 (for the second program)
```

6.3.5 Data Types

Data types in C refer to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.

Slno	Description of data type
1	Basic : they are arithmetic types and are further classified into a) integer b) floating point
2	Enumerated: they are again arithmetic types and they are used to define variables that can only assign certain discrete integer values throughout the program
3	They type void: the type specifier void indicates that no value is availabe
4	Derived types: they include a) pointer b) arrays c) structures d) union e) functions

Now we will learn all the data types in detail:

Integer:

The following table provides the details of standard integer types with their storage size and value ranges:

Type	Storage size	Value range
Char	1 byte	-128 to 127 or 0 to 255
Unsigned char	1 byte	0 to 255
Signed char	1 byte	-128 to 127
Int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
Unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
Short	2 bytes	-32768 to 32767
Unsigned short	2 bytes	0 to 65535
Long	4 bytes	2,147,483,648 to 2,147,483,647

Unsigned long	4 bytes	0 to 4,294,967,295

To get the exact size of a type or a variable on a particular platform, you can use the sizeof operator. The expressions sizeof(type) yields the storage size of the object or type in bytes.

Floating point:

The following table provides the details of standard floating-point types with storage sizes and value ranges and their precision:

Type	Storage size	Value range	Precision
Float	4 bytes	1.2E-38 to 3.4E+38	6 decimal places
Double	8 bytes	2.3E-308 to 1.7E+308	15 decimal places
Long double	10 bytes	3.4E-4932 to 1.1E+4932	19 decimal places

The void type:

The void type specifies that no value is available. It is used in three kinds of situations which is as follows:

Sno.	**Types and description**
1	Function returns as void There are various functions in C which do not return any value or you can say they return void. A function with no return value has the return type as void. For example: **void exit (int status);**
2	Function arguments as void There are various functions in C which do not accept any parameter. A function with no parameter can accept a void. For example: **int rand(void);**
3	Pointers to void A pointer of type void * represents the address of an object, but not its type. For example, a memory allocation function **void *malloc(size_t size);** returns a pointer to void which can be casted to any data type.

6.4 Operators and Expressions

An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators:

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment operator
- Increment and Decrement operators
- Conditional Operator
- Bitwise Operators and special operators
- Arithmetic Expression

6.4.1 Arithmetic Operator

The following are arithmetic (mathematical) operators used to construct arithmetic expressions.

Operator	Meaning	Example	
		Expression	Value
+	Addition	9+5	14
-	Subtraction	9-5	4
*	Multiplication	9*5	45
/	Division	9/5	1 (Integer) 1.8 (Real)
%	Modulo (remainder when first number divides second number)	9%5	4

Can you construct an expression to add two variables say x and y ?

Consider a small example that implements Arithmetic operation

```
#include <stdio.h>

main()
{
  int a = 21;
  int b = 10;
  int c ;

  c = a + b;
printf("Line 1 - Value of c is %d\n", c );
  c = a - b;
printf("Line 2 - Value of c is %d\n", c );
  c = a * b;
printf("Line 3 - Value of c is %d\n", c );
  c = a / b;
printf("Line 4 - Value of c is %d\n", c );
  c = a % b;
printf("Line 5 - Value of c is %d\n", c );
  c = a++;
printf("Line 6 - Value of c is %d\n", c );
  c = a--;
printf("Line 7 - Value of c is %d\n", c );

}
```

6.4.2 Relational Operator

We often compare two quantities, and depending on their relation, take certain decisions. For example, we may compare the age of two persons, or the price of two items, and so on. These comparisons can be done with the help of relational operators. An expression such as a < b or 1 < 20 containing a relational operator is termed as a relational expression. The value of a relational expression is either one or zero. It is one if the specified relation is true and zero if the relation is false. For example

10 < 20 is true and hence the value is one

but

20 < 10 is false and hence the value is zero

The relational operators and their meanings are shown below.

Operator	Meaning
<	is less than

<=	is less than or equal to
>	is greater than
>=	is greater than or equal to
==	is equal to
!=	is not equal to

(4.5 <= 10) is TRUE

(4.5 >-10) is FALSE

(-35 >= 0) is FALSE

(10 < 7+5) is TRUE

((a+b) == (c+d)) is TRUE, only if the sum of values of a and b is equal to the sum of values of c and d.

Relational expressions are used in decision making statements such as, “if”, “while” statements etc., to decide the course of action of a running program.

Can you construct a relational expressing for checking whether your age is greater than 25?

6.4.3 Logical Operators

In addition to the relational operators, C has the following three logical operators.

&&	meaning	logical AND
||	meaning	logical OR
!	meaning	logical NOT

The logical operators && and || are used when we want to test more than one condition and make decisions. An example is

((age> 28) && (designation = =”Programmer”))

An expression of this kind which combines two or more relational expressions is termed as a logical expression or a compound relational expression. Like the simple relational expressions, a logical expression also yields a value of one or zero, according to the truth table shown in the table below. The logical expression given above is true only if age >28 is true and designation is equal to "Programmer" is true. If either or both of them are false, the expression is false.

Condition1	**Condition 2**	**AND (&&)**	**OR (‖)**	**NOT (!)**	
				For condition1	For condition2
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	False
False	False	False	False	True	True

Some examples of the usage of logical expressions are:

1. ((number > 0) && (number < 100))

The above condition is true only when the number takes the value between 0 and 100.

2. ((word_category = "noun") || (word _pos = 5))

The above condition is true only when 5th word is noun.

Now can you construct a logical expression to check

a) whether the given character is a vowel or not

b) whether the given number is an even number greater than 20

6.4.4 Assignment Operator

The following table lists the assignment operators supported by the C language –

Operator	**Description**	**example**

=	Simple assignment operator. Assigns values from right side operands to left side operand	C = A + B will assign the value of
+=	Add AND assignment operator. It adds the right operand to the left operand and assign the result to the left operand.	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator. It subtracts the right operand from the left operand and assigns the result to the left operand.	C -= A is equivalent to C = C – A
*=	Multiply AND assignment operator. It multiplies the right operand with the left operand and assigns the result to the left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator. It divides the left operand with the right operand and assigns the result to the left operand.	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator. It takes modulus using two operands and assigns the result to the left operand.	C %= A is equivalent to C = C % A
<<=	Left shift AND assignment operator	C <<= 2 is same as C = C << 2
>>=	Right shift AND assignment operator	C >>= 2 is same as C = C >> 2
&=	Bitwise AND assignment operator	C &= 2 is same as C=C & 2
^=	Bitwise exclusive OR and assignment operator	C ^= 2 is same as C = C ^ 2

|=	Bitwise inclusive OR and assignment operator	C |= 2 is same as C = C | 2

6.4.5 Increment and Decrement Operator

Increment and decrement operator are unary operators. Unary operators require only one operand. Depending on the operator use, the value of the expression changes. For example, see the table given below.

Operator	Meaning	Example	
		Expression	Value
+	Signed (positive)	+5	5
-	Singed (negative)	-5	-5
&	Address of (always used with the variables)	&a	Address of a
*	Pointer (always used with the variables)	*a	the value stored in a

Increment operators are used to increase the value of the variable by one and decrement operators are used to decrease the value of the variable by one in C programs.

Syntax:

Increment operator: ++var_name; (or) var_name++;
Decrement operator: **- -**var_name; (or) var_name - -;
Examples:
Increment operator: ++i; i++;
Decrement operator: - -i; i- -;

Program to perform increment using increment operator

```
#include <stdio.h>
int main()
{
   int i=1;
   while(i<10)
   {
```

```
printf("%d ",i);
i++;
    }
}
```

```
Output: 1 2 3 4 5 6 7  8 9
```

Prefix &postfix Operators

The traditional method of incrementing numbers is as follows.

```
        a = a + 1;
```

In C, this syntax is valid but you can also use the ++ operator to perform the same function.

```
        a++;
```

will also add 1 to the value of **a**. By using a similar syntax you can also decrement a variable as shown below.

```
        a--;
```

These operators can be placed as a postfix or prefix as below:

```
        a++;                ++a;
```

When used on as above the post fix or prefix have the same effect but within an expression there is a subtle difference.

1. Prefix notation will increment the variable before the expression is evaluated.
2. Postfix notation will increment after the expression evaluation.

Here is an example-1.

```
main()
{
 int a=1;
printf(" a is %d", a++);
     }
```

Here is an example-2.

```
main()
{
 int a=1;
printf(" a is %d", ++a);
        }
```

In both examples, the final value of **a** will be 2. But the first example will print 1 and the second will print 2. Can you guess why?

6.4.6 Conditional Operator

- The conditional expression can be used as shorthand for some if-else statements.
- Conditional operator is also called as ternary operator.
- This operator consist of two symbols: the question mark (?) and the colon (:).

The general syntax of the conditional operator is:
Identifier = (test expression)? Expression1: Expression2 ;

This is an expression, not a statement, so it represents a value. The operator works by evaluating test expression. If it is true (non-zero), it evaluates and returns expression1. Otherwise, it evaluates and returns expression2

6.4.7 Bit wise and special operators

Bitwise operators operate on individual bits of integer (int and long) values. This is useful for writing low-level hardware or OS code where the ordinary abstractions of numbers, characters, pointers, and so on, are insufficient. Bit manipulation code tends to be less “portable”. Code is “portable” if without any programmer intervention it compiles and runs correctly on different types of computers. The bit wise operations are commonly used with unsigned types. These operators perform bit wise logical operations on values. Both operands must be of the same type and width: the resultant value will also be this type and width.

Biwise operators are:

~ Bitwise Negation one’s complement (unary)
& Bitwise And
| Bitwise Or
^ Bitwise Exclusive Or
>> Right Shift by right hand side (RHS) (divide by power of 2)
<< Left Shift by RHS (multiply by power of 2)

Definition of bitwise logical operators:

Bit a	Bit b	a & b	a | b	a ^ b	~a
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Example:

Bitwise Negation
a=00000011
~a =11111100

Bitwise And
a = 00000011
b = 00010110
a&b=00000010

Bitwise Or
a = 00000011
b = 00010110
a|b=00010111

Bitwise Exclusive Or
a = 00000011
b = 00010110
a^b=00010101

Right Shift

a = 0000 0011 = 3

(a<<=1) = 00000110 = 6

(a<<=2) = 00011000 = 24

(a<<=3) = 11000000 = 192

Left Shift

a=11000000 =192

(a>>=1) = 01100000 = 96

(a>>=2) = 00011000 = 24

(a>>=3) = 0000 0011 = 3

6.4.8 Arithmetic Expression

C has a wide range of operators. An arithmetic expression is composed of operators and operands. Operators act on operands to yield a result. Commonly used arithmetic operators are +, -, *, / and %.

The plus sign (+) is used to add two values, the minus sign (-) to subtract one value from another, the asterisk(*) to multiply two values, the division (/) to divide a value and the modulus (%) to obtain the reminder of integer division. These are known as binary operators since they operate on two values or variables.

Following are examples of arithmetic expressions :

```
result = x - y;
total = principle + interest;
numsquare = x * x;
celcius = (fahrenheit - 32) / 1.8
```

Notice the equal sign (=) in the above expressions, it is known as the assignment operator. It assigns the value on the right hand side of the equal sign to the variable on the left hand side. In the last expression, parentheses are used to perform a certain operation first. This is because in C, operators follow a precedence rule. *, / and % have a higher precedence over + and -.

6.5 Summary

1. C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs.
2. C language is case sensitive language.
3. Every statement in C should end semicolon (;).
4. A C program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.
5. A variable definition tells the compiler where and how much storage to create for the variable.
6. Constants refer to fixed values that the program may not alter during its execution. These fixed values are also called literals.
7. Constants can be of any of the basic data types like an integer constant, a floating constant, a character constant, or a string literal.
8. Data types in C refer to an extensive system used for declaring variables or functions of different types.
9. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.
10. An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators:

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators

6.6 SQA

1. Write a short note on different types constant in C?
2. Write a short note on C data types?
3. Write a short note on C Operators?
4. Briefly explain the operator precedence?

5 mark questions

1. How do you construct variable names in C

2. What are white space characters and how they are represented?
3. What are different types of string Constants? Explain them with examples
4. Write a short note on relational operators?
5. What do you know about Logical operators
6. What are prefix and post operators in C

Objective Type questions

1. 1. 123 is a -------------- constant (Integer)
2. 2. ‘X’ is a ------------ constant (single quoted string constant)
3. Single quoted string can have ----- value (Integer)
4. 4. The value of ---------- changes during the execution of the program (variable)
5. 5. The value of ---------- does not change during the execution of the program (constant)
6. 6. The data type used to return null values ------------ (void)
7. ------ data type is used to store real numbers. (float)
8. ---- is the only special character used in variable names. (underscore)
9. 1. 4%3 is ------- (1)
3. Relational expression to check whether a and b are equal ---------- (a==b)
4. 2*3/2 is ------ (3)
5. (2*3)/2 is ------- (3)
6. 2*(3/2) is ------ (2)
7. The address operator is ----- (&)
8. The pointer operator is ----- (*)
9. ++ is known as ----- (Auto increment Operator)
10. --- is known as ------ (Auto decrement Operator)
10. 10. Construct an expression to find out all vyakaranam teachers whose salary is greater than 30,000 [((subject ==”vyakaranam) &&(salary >=30000))]

Unit – VII Decision making statements

Structure

7.0 Objective

The main objective of this unit is to learn about decision making statements in C language.

7.1 Introduction

The C language is accompanied by a collection of library functions which includes a number of input/output functions.These functions are used to permit the transfer of information between the computer and the standard input/output device. The basic input/output functions aregetchar, putchar, puts, scanf and printf. The first two functions, getchar and putchar, are used to transfer single characters. The next function puts is used to output strings, and the last two functions, scanf and printf, permit the transfer of single characters, numerical values and strings.

7.2 Input and Output Operations

Input means to provide the program with some data to be used in the program.

Output means to display data on screen or write the data to a printer or a file.

C programming language provides many built-in functions to read any given input and to display data on screen when there is a need to output the result.

7.2.1 Reading and Writing a character

The getchar() and putchar() Functions:

The **int getchar(void)** function reads the next available character from the screen and returns it as an integer. This function reads only single character at a time. You can use this method in the loop in case you want to read more than one character from the screen.

The **int putchar(int c)** function puts the passed character on the screen and returns the same character. This function puts only single character at a time. You can use this method in the loop in case you want to display more than one character on the screen. Check the following example −

```
#include <stdio.h>

#include <ctype.h>

int main()

{

  char c;

printf("Enter some character. Enter $ to exit...\n");

  while (c != '$');

  {

    c = getchar();

printf("\n Entered character is: ");

putchar(c);

printf("\n")

  }
  return 0;
}
```

Output :

```
Enter some character. Enter $ to exit…
A
Entered character is :A
B
$
Entered character is: $
```

The gets() and puts() Functions

The **char *gets(char *s)** function reads a line from **stdin** into the buffer pointed to by **s**until either a terminating newline or EOF (End of File).

The **int puts(const char *s)** function writes the string 's' and 'a' trailing newline to **stdout**.

scanf() and printf() or input and output statements

Input statement : As the name suggests, the input statement is used to get input from the user into the computer. This input is stored in the computer into a memory location. As already told, we can store numeric and string values (strings having single andmulti characters).
The simple statement used to read in C is scanf. The general format of scanf is as follows:

```
scanf("format specifier", &variable1, &variable2,....);
```

The format specifier contains the format of data being received. The ampersand symbol, "&" before each variable name is an operator that specifies the variable name's address. We must always use this operator, otherwise unexpected results may occur. Let us look at an example:

```
scanf("%d", &a);
```

When this statement is encountered by the computer, the execution stops and waits for the value of the variable name to be typed in. Since the format specifier "%d" specifies that an integer value is to be read from the terminal, we have to type in the value in integer form. Once the number is typed in and the 'Return' Key is pressed, the value is stored in the memory location named as “a” and the program execution then proceeds to the next statement. Thus, the use of scanf provides an interactive feature and makes the program 'user friendly'.

Can you write a statement to read an integer value into a variable called temp?

Output statement: The printf is used to print the literal (string or textual) message and the contents of memory locations. The printf function is usually of the form:

```
printf(("format specifier",variable,variable,variable...)
```

You can carry on writing a list of variables separated by commas as long as you want to. The format specifier is all-important because it specifies the type of each variable in the list and how you want it printed. The printf scans the string from left to right and prints on the screen, or any suitable output device, any characters it encounters - except when it reaches a % character. The % character is a signal that what follows it is a specification for how the next variable in the list of variables should be printed. The function printf uses this information to convert and format the value that was passed to it. For example:

```
printf("Hello World");
```

only has a string and, as this contains no % characters it results in Hello World being displayed and doesn't need to display any variable values. The specifier %d means convert the next value to a signed decimal integer and so:

```
printf("Total = %d",total);
```

will print “*Total =*” and then the value passed by total as an integer.

Can you print a variable called sum as an integer along with some message (like “The sum of given numbers is ----“)?

Format specifiers

Some of the most commonly used format specifiers are given below:

'%d' : Scan an integer as a signed decimal number.
'%f' : Scan a floating-point number in normal (fixed-point) notation.
'%c' : Scan a character (char).

Header Files

It would be very tedious, for all of us, if every time we wanted to communicate with the computer we all had to write our own output functions. Fortunately, we do not have to. C uses libraries of standard functions which are included when we build our programs. So all our C programs will need at least one of C's standard libraries which deals with standard inputting and outputting of data. This library is called stdio.h and it is declared in our programs before the main function, when we use input and output statements. The .h extension indicates that this is a header file.
All pre-processor directives begin with a # and the must start in the first column. The commonest directive to all C programs is:

```
#include <stdio.h>
```

Note the use of the angle brackets (< and >) around the header's name. These brackets indicate that the header file is to be looked for, on the system disk which stores the C software. Some text books will show the above statement as follows:

```
#include "stdio.h"
```

The double quotes indicate that the current working directory should be searched for the required header file. This will be true when you write your own header files but the standard header files should always have the angle brackets around them.

List of some Header files

<ctype.h>	This header file contains functions used to classify characters by their types or to convert between upper and lower case in a way that is independent of the used character set
<errno.h>	For testing error codes reported by library functions.
<float.h>	Contains defined constants specifying the implementation-specific properties of the floating-point library, such as the minimum difference between two different floating-point numbers (_EPSILON), the maximum number of digits of accuracy (_DIG) and the range of numbers which can be represented (_MIN, _MAX).
<inttypes.h>	For precise conversion between integer types.
<limits.h>	Contains defined constants specifying the implementation-specific properties of the integer types, such as the range of numbers which can be represented (_MIN, _MAX).
<locale.h>	For setlocale() and related constants. This is used to choose an appropriate locale.
<math.h>	For computing common mathematical functions

<stdarg.h>	For accessing a varying number of arguments passed to functions.
<stdbool.h>	For a boolean data type
<stdint.h>	For defining various integer types.
<stdio.h>	Provides the core input and output capabilities of the C language. This file includes the venerable printf function.
<stdlib.h>	For performing a variety of operations, including conversion, pseudo-random numbers, memory allocation, process control, environment, signalling, searching, and sorting.
<string.h>	For manipulating several kinds of strings.
<time.h>	For converting between various time and date formats.

Sample C programs

Now we will try to write some primitive programs in C.
Ex. 1. Write a program in C to add two numbers.

Sl. No.	Algorithm	Corresponding C instruction
1	read a,b	scanf("%d,%d" &a,&b);
2	c = a+b	c = a+b;
3	print a,b,c	printf("The sum of %d and %d is %d \n", a,b, c);

To be more clear, the above program may be further be modified as

C program	Explanation
/* Program#scanf.c A Simple Program to Demonstrate scanf */	This a comment statement. This is inserted anywhere in the program to increase the understandability of the users of the program. However, this is not executed by the compiler.
#include <stdio.h>	This is the header file name which tells the compiles that the functions scanf and printf are defined in stdio.h
main()	Name of the main C program
{	Indicates the beginning of main of C program
int a,b,c;	Before using any variables in the program, they must be declared along with their data type
printf("\nThe first number is ");	Displays the message to the user to enter in some value for a
scanf("%d",&a);	Expects a value to read in to a variable a
printf("The second number is ");	Displays the message to the user to enter in some value for a
scanf("%d",&b);	Expects a value to read in to a variable a
c=a+b;	A and B are added and stored in C
printf("The answer is %d \n",c);	The calculated result is printed with an appropriate message
}	Indicates the end of the main program

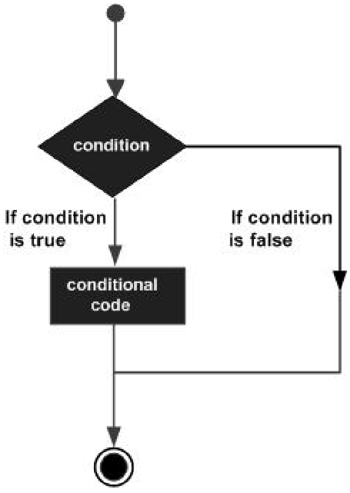
Ex. 2 Write a program in to calculate the area of a rectangle.

```
#include <stdio.h>
```

```
main()
{
/* calculates the area of rectangle*/
int length, breadth, area;
printf("Enter the length and  Breadth\n");
scanf("%d %d\n" &length, &breadth);
area = length * breadth;
printf("The are of the given rectangle %d \n", area);
}
```

7.3 Decision making and Branching

We often take decisions depending on some conditions. The condition, after evaluation takes one of two values, viz., "true" or "false". Depending on the value of the condition, the control is to be transferred to appropriate statement. Show below is the general form of a typical decision making structure found in most of the programming languages −



C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value.

7.3.1 If statement

The general format of a simple if statement is

```
    if (condition)
```

```
{
      statement-block;
}
statement-n;
```

The 'statement-block' may be a single statement or a group of statements. If the condition is true, the statement-block will be executed; otherwise the statement-block will be skipped and the execution will jump to the statement-n. Remember, when the condition is true both the statement-block and the statement-n are executed in sequence.

Consider the following segment of a program.

```
if (a>b)
     {
       printf("%d is greater than %d \n", a,b);
      }
```

The program tests the value of a with the value of b. If a is greater than b, it print that a is greater than b. Can you construct an if statement to check whether the given number is equal to 200 ?

7.3.2 if..else statement

The if…else statement is an extension of the simple if statement. The general form is

```
if (condition)
{
        statement-block-1
        }
 else
{
        statement-block-2
}
statement-n;
```

If the condition is true, then the statement-block-1 (the series of statements enclosed by flower brackets, immediately following the if statement are executed; otherwise, the statement-block-2 are executed. In either case, either statement-block1 or statement-block2 will be executed, not both, in both the cases, the control is transferred subsequently to statement-n. Consider the following segment of a program.

```
if (a>b)
{
printf("%d is greater than %d \n", a,b);
}
else
{
printf("%d is greater than %d \n", b,a);
}
```

Now write an if-else statement to print "the student passes the exam" if theaverage_marks is greater than 40 otherwise print "the student fails".

7.3.3 Nesting of if …else

It is always legal in C programming to **nest** if-else statements, which means you can use one if or else if statement inside another if or else if statement(s).

Syntax

```
if (condition)
//nested if else inside the body of "if"
If(condition2)
{
 //statements inside the body of nested "if"
}
else  {
```

```
//statements inside the body of nested “else”

}

}
else   {
//statements inside the body of  “else”
}
```

Consider the following example for nested if

```
#include<stdio.h>

main()
{
int x=20,y=30;

if(x==20)
{
if(y==30)
{
printf("value of x is 20, and value of y is 30.");
}
}
}
```

Output:
(try the output)

7.3.4 Switch statement

When one alternatives out of two is to be selected, if statement can be used to control the selection. However, the complexity of such a program increases when the number of alternatives increases. The program becomes difficult to read and follow. At times, it may confuse even the person who designed it. Fortunately, C has a built-in multi way decision statement know as a switch. The switch statement tests the value of a given variable (or expression) against a list of case values and when a match is found, a block of statements associated with the case is executed. The general form of the switch statement is as shown below:

```
switch (expression)
{
        case value-1:
           block-1
        break;
        case value-2:
           block-2
        break;
          ……
          ……
        default
           default-block
        break;
}
statement-x;
```

The expression is an integer expression or characters. value-1, value-2 …are constant or constant expressions and are known as case labels. Each of these values should be unique within a switch statement. Block-1, block-2 … are statement lists and may contain zero or more number of statements. There is no need to put braces around these blocks. Note that case labels end with a colon (:).

When the switch is executed, the value of the expression is successively compared against the values value-1, value-2, …. If a case is found whose value matches with the value of the expression, then the block of statements that follows the case are executed.

The break statement at the end of each block signals the end of a particular case and causes and exit from the switch statement, transferring the control to the statement-x following the switch.

The default is an optional case. When present, it will be executed if the value of the expression does not match with any of the case values. If not present, no action takes place if all matches fail and the control goes to the statement-x.
The switch statement can be used to convert the given single digit into a corresponding number.

```
#include <stdio.h>

main()
{
int number;

/* read number*?
scanf("%d", &number);

/* print the given digit into corresponding word */
 switch(number)
{

        case 0 :
printf("zero\n");
                break;
        case 1 :
printf("One\n");
                break;
        case 2 :
printf("Two\n");
                break;
        case 3 :
printf("Three\n");
                break;
        case 4 :
printf("Fourn");
                break;
        case 5 :
printf("Five\n");
                break;
        case 6 :
printf("Six\n");
                break;
        case 7 :
printf("Seven\n");
                break;
        case 8 :
printf("Eight\n");
                break;

        case 9 :
printf("Nine\n");
                break;
        case 10 :
printf("Ten\n");
                break;
        default
printf("Number  is not a single digit number\n”;
               break;
}
}
```

Now can you write a program to print the grades in the following way.

A- Excellent
B – Very Good
C- Good
D – Fair
E - Fail

7.3.5 Go to Statement

C also provides a goto statement which permits unstructured jumps to be made.However, its use is not recommended.

The goto statement transfers program execution to some label within the program.

The syntax of goto statement is given below:

```
goto label;
....
label:
statement-block;
```

The example of goto is shown below:

```
gotonext_block;
printf("The statement is  not executed.\n");

next_block:
printf("This statement is executed!\n");
```

7.4 Summary

- Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.
- **If statement:** An if statement consists of a Boolean expression followed by one or more statements

- **If…else statement:** An if statement can be followed by an optional else statement, which executes when the Boolean expression is false.

- **Nested if statement:** One if or else if statement inside if or else if statement(s).

- **Switch statement:** A switch statement allows a variable to be tested for equality against a list of values.

7.5 SAQ

10 mark Questions

1. Explain the syntax of if..else statement with suitable example
2. Explain the syntax of switch statement with suitable example

5 mark question

1. Write a short note on goto statement with examples
2. Write a short note on break statement.
3. Write a C program to check whether two given numbers are equal or not.
4. Write a C program to check whether two given numbers are equal or not

1 mark question

1. The condition when executed will have ------ values. (2)
2. The value of condition 5>2 is ------ (True)
3. The value of year>2010 is ----- (False)
4. The break statement causes ------ from the loop. (Exit)
5. The optional case in switch statement is ------ (default)
6. Simple if is used when ---- alternative are to be used. (one)
7. If..else is used when ----- out of two alternatives are to be used. (one)
8. Switch statement is used when --- out of many alternatives are used (One)
9. The values to be compared in switch statement are indicated by --- (case)
10. case labels end with ---- (:)
11.---- is an optional case in switch statement (default)

Unit – VIII Decision making and looping

Structure

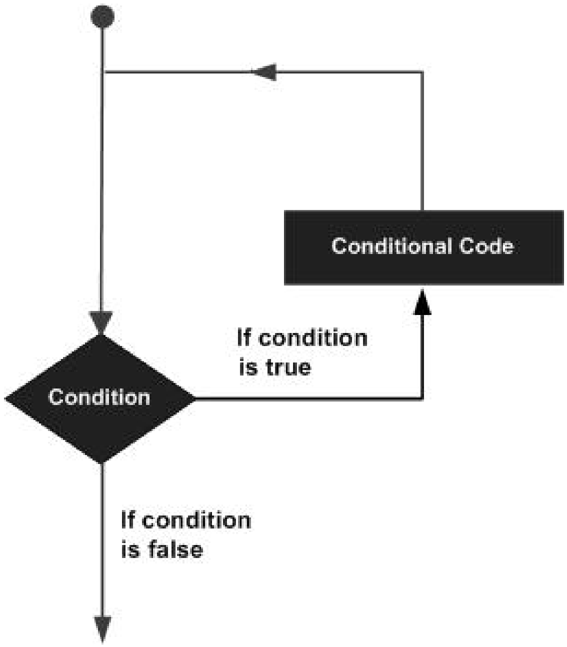
8.0 Objective

The main objective of this unit is to make students clearly understand what are decision making and looping statements in C

8.1 Introduction

C gives a variety of loops so that we can write the same segment of C code in many alternative ways. These alternative ways make the C language user-friendly. The loops offered by C are discussed in detail in the following sub sections.

The loop is a common element in all programming languages. Loops allow you to execute a sequence of statements repeatedly, using a control variable and a keyword like for, while etc. The value of a control variable, which is really just a counter, determines how many times a statement or a statement-block is to be executed. The value of the counter is set, activated and controlled by the keyword



8.2 While Statement

The simplest of all the looping structures in C is the while statement. The basic format of the while statement is

```
while (condition)
{
        body of the loop
}
```

The while is an entry-controlled loop statement. The condition is evaluated and if the condition is true, then the body of the loop is executed. After execution of the body, the test-condition is once again evaluated and it if is true, the body is executed once again. This process of repeated execution of the body continues until the test-condition finally becomes false and the control is transferred out of the loop. On exit from the loop, the program continues with the statement immediately after the body of the loop.

The body of the loop may have one or more statements. The braces are needed only if the body contains two or more statements. However, it is a good practice to use braces even if the body has only one statement.

The following is an example for while loop to calculate the sum of first 50 numbers.

```
 int i = 1, sum = 0;
 while (i<= 50)
 {
   sum += i;
i++;
 }
printf (“The sum of first  50 numbers is %d\n”, sum);
```

The body of the loop is executed 10 times for i = 1,2, …, 50 each time adding the value of i to sum, which is incremented inside the loop. The test condition may also be written as i< 51; the result would be the same.

Now can you write a while loop statement to add first 10 even numbers?

8.3 Do ..While statement

The syntax of do-while is given below.

```
  do
 {
         body of the loop
  }
 while (condition);
```

on reaching the do statement, the program proceeds to evaluate the body of the loop first. At the end of the loop, the condition in the while statement is evaluated. If the condition is true, the program continues to evaluate the body of the loop once again. This process continues as long as the condition is true. When the condition becomes false, the loop will be terminated and the control goes to the statement that appears immediately after the while statement.

Since the condition is evaluated at the bottom of the loop, the do…while construct provides an exit-controlled loop and therefore the body of the loop is always executed at least once.

A simple example of a do… while loop is:

```
do
{
        printf("Input a number\n");
        number = getnum();
}
while (number > 0);
```

This segment of a program reads a number from the keyboard until a zero or a negative number is keyed in.

Now can you write a do..while loop statement to add first 10 even numbers?

8.4 for statement

The "for" loop is another entry-controlled loop that provides a more concise loop control structure. The general form of the for loop is

```
for (initialization ; condition; increment)
{
        body of the loop
}
```

The execution of the for statement is as follows:

1. Initialization of the control variables is done first, using assignment statements such as i=1 and count = 0. the variables i and count are known as loop-control variables.
2. The value of the control variable is tested using the test-condition. The test-condition is a relational expression, such as i< 10 that determines when the loop will exit. If the condition is true, the body of the loop is executed; otherwise the loop is terminated and the execution continues with the statement that immediately follows the loop.
3. When the body of the loop is executed, the control is transferred back to the for statement after evaluating the last statement in the loop.

Now, the control variables is incremented using an assignment statement such as i = i+1 and the new value of the control variable is again tested to see whether it satisfies the loop condition. If the condition is satisfied, the body of the loop is again executed. This process continues till the value of the control variable fails to satisfy the test-condition.

```
int sum = 0;
for (int i = 1; i<= 50; i++)
{
  sum += i;
}
```

Now can you write a for loop statement to add first 10 even numbers?

8.5 Summary

- A loop statement allows us to execute a statement or group of statements multiple times.
- While loop: Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
- Do…while loop: It is more like a while statement, except that it tests the condition at the end of the loop body.
- For loop: Executes a sequence of statements multiple times and abbreviates the code that manages the loop variables.

8.6 SAQ

10 marks questions

1. Explain while statement with examples

2. Explain for statement with examples

3. Explain Do..while statement with examples

5 marks questions

Write a program in C using for, while ,do..while statements

a) to print the sum of first 25 given numbers

b) to print the average marks obtained by the students

c) to accept two numbers and to print the odd numbers between them

d) to print the multiplicative table of a given number

e) to keep on asking numbers until '5' is entered. It will then display the number of times a number was entered and display its sum.

1 mark questions

1) The while is an -------- controlled loop statement (entry)

2) The do…while construct provides an ------- controlled loop (exit)

3) ----- controls the for loop. (initialization variable)

Unit – IX Arrays

Structure

9.0 Objective

9.0 Objective

In this unit students will make clear about what are arrays? Different types of arrays and its implementation. Here they will learn about strings in detail like what are string and how to handle string functions.

9.1 Introduction

An array is a group of related data items that share a common name. For instance, we can define an array of names to represent a group of employees. A particular name is indicated by writing a number called index number or subscript in brackets after the array name. For example,

```
name[10]
```

represents the name of the 10th employee. While the complete set of names is referred to as an array called "name", the individual values are called elements viz., name[0], name[1] .. name[10] etc. Arrays can be of any variable type.

9.1.1 One Dimensional Arrays

A list of items can be given one variable name using only one subscript and such a variable is called a single-subscripted variable or a one-dimensional array.

For example, if we want to represent a set of four numbers, then we may declare the array as follows.

```
char number[4];
```

five storage locations are reserved as shown below:

number[0]
number[1]
number[2]
number[3]
number[4]

The values to the array elements can be assigned as follows:

number[0] = 35;

number[1] = 40;

number[2] = 20;

number[3] = 57;

number[4] = 19;

This would cause the array number to store the values as shown below:

35
40

20	number[0]
57	number[1]
19	number[2]
	number[3]

number[4]

These elements may be used in programs just like any other C variable. For example, the following are valid statements:

Statement	Functionality	Values of number array
a = number[0] + 10;	10 is added to the first element of number array and the result is stored in a . Hence a becomes 45	number [0] = 35
number[4] = number[0] + number[2];	The first and third elements of number array are added and the result is stored in the fifth elelement	number [4] = 55
number[2] = x[5] + y[10];	The 6th element of x and 11th element of y are added and the result is stored in number [2]	If x[5]1=10, y[10] = 11, then number [2] =21
value[6] = number[i] * 3;	The ith element of number is multiplied by 3 and the result is stored as value[6]	If i =2, value[6] becomes 60

The subscript of an array can be integer constants, integer variables like I, or expressions that yield integers. C performs no bounds checking and, therefore, care should be exercised to ensure that the array indices are within the declared limits.

Declaration of Arrays

Like any other variable, arrays must be declared before they are used. The general form of array declaration is

```
type variable-name[size];
```

The type specifies the type of element that will be contained in the array, such as int, float, or char and the size indicates the maximum number of elements that can store inside the array. For example,

```
float height[50];
```

Declares the height to be an array containing 50 real elements. Any subscripts 0 to 49 are valid.

Initialization of Arrays

We can initialize the elements of arrays in the same way as the ordinary variables when they are declared. The general form of initialization of arrays is:

```
type array-name[size] = {list of values};
```

The values in the list are separated by commas. For example, the statement

```
int number[3] {0,0,0};
```

Will declare the variable number as an array of size 2 and will assign zero to each element. If the number of values in the list is less than the number of elements, then only that many elements will be initialized. The remaining elements will be set to zero automatically. For instance

```
float total[5] = {0.0,15.75,-10};
```

Will initialize the first three elements to 0.0,15.75, and –10.0 and the remaining two element to zero.

Reading One Dimensional Arrays

To read an one dimensional array of size 5, see the following example .

```
 for (index = 0; index < 5; ++index)
 {
scanf(“%d “, &array[index]);
          }
```

Now can you write C code to read a name of 10 characters of maximum size?

Printing One Dimensional Arrays

To print an one dimensional array of size 5, see the following example .

```
for (index = 0; index < 5; ++index)
 {
printf("the %d element of array is %d\n”, index, array[index] );
   }
```

Now write a C statement to print an array of student heights containing a name of 10 elements?

DECLARING AND INITIALIZING STRING VARIABLES

A string variable is any valid C variable name and is always declared as an array. The general form of declaration of a string variable is

```
char string_name[size];
```

The size determines the number of character in the string-name. Some examples are:

```
char city[10];

char name[30];
```

When the compiler assigns a character string to a character array, is automatically supplies a null character (‘\0’) at the end of the string. Therefore, the size should be equal to the maximum number of characters in the string plus one.

Character arrays may be initialized when they are declared. C permits a character array to be initialized in either of the following two forms:

```
static char city[9] = “NEW YORK”;

static char city[9] = {‘N’, ‘E’, ‘W’, ‘’, ‘Y’, ‘O’, ‘R’,
‘K’,’\0’};
```

WRITING STRINGS TO SCREEN

We have used extensively the printf function with %s format to print strings to the screen. The format %s can be used to display an array of characters that is terminated by the null character. For example, the statement

```
printf(“%s”, name);
```

Can be used to display the entire contents of the array name.

We can also specify the precision with which the array is displayed. For instance, the specification

```
%10.4
```

indicates that the first four characters are to be printed in a field width of 10 columns.

Write a C program to read and print a list of student names.

9.1.2 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. Consider the following data table, which shows the value of marks of three students:

	Mark1	Mark2	Mark3
Student1	50	45	65
Student2	60	55	55
Student3	70	65	45

The table contains a total of 12 values, three in each line. We can thin of this table as a matrix consisting of four rows and three columns. Each row represents the values of marks by a particular student and each column represents the values of marks of a particular subject.

INITIALIZING TWO-DIMENSIONAL ARRAYS

Like the one-dimensional arrays, two-dimensional arrays may be initialized by following their declaration with a list of initial values enclosed in braces. For example

```
int table[3][3] = {50,45,65,60,55,55,70,65,45};
```

Initializes the elements of the first row to zero and the second row to one. The initialization is done row by row. The above statement can be equivalently written as

```
int table[3][3] = {{50,45,65}, {60,55,55},{70,65,45}};
```

by surrounding the elements of each row by braces.

We can also initialize a two-dimensional array in the form of a matrix as shown below:

```
int table[3][3] = {
            {50,45,65},
                                    {60,55,55},
                                    {70,65,45}
      };
```

Reading two Dimensional Arrays

To read an two dimensional array of names (of maximum length 10 characters) of 6 students in a class, the following C program segment is given below

```
for (int i = 0; i< 10; ++i)
{
   for (int j = 0; j < 6; ++j)
   {
      scanf("%c", &student_name[i][j]);
   }
scanf("%\n");
}
```

Can you explain how the above segment works?

Printing two Dimensional Arrays

To print an array of names (of maximum length 10 characters) of 6 students of a class, the following C program segment is given below

```
for (int i = 0; i< 10; ++i)
{
   for (int j = 0; j < 6; ++j)
   {
      printf("%c", student_name[i][j]);
   }
printf("\n");
}
```

Try to take an example data to see how the above C code is executed?

9.2 Strings

Strings are actually one-dimensional array of characters terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null. The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

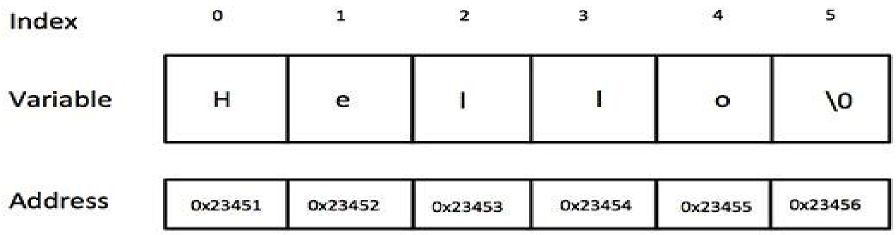
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

If you follow the rule of array initialization, then you can write the above statement as follows:

char greeting[] = "Hello";

Following is the memory presentation of the above defined string in C/C++:

Actually, you do not place the null character at the end of a string constant. The C compiler automatically places the '\0' at the end of the string when it initializes the array. Let us try to print the above mentioned string:



Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

Actually, you do not place the *null* character at the end of a string constant. The C compiler automatically places the '\0' at the end of the string when it initializes the array.

```
nclude<stdio.h>

int main ()
{
  char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

printf("Greeting message: %s\n", greeting );

  return 0;
}
```

Output:

```
Greeting message: Hello
```

9.3 String Handling Function

Character strings are often used to build meaningful and readable programs. The common operations performed on character string are:

- Reading and Writing string.
- Combining strings together.
- Copying are string to another.
- Comparing strings for equality.
- Extracting a portion of a string.

Fortunately, the C library supports a large number of string-handling functions that can be used to carry out many of the string manipulations discussed so far.

How to read and write strings are discussed in the last chapter. Now let us see how other tasks can be done.

Following are the most commonly used string-handling functions.

Function	**Action**
strcat()	concatenates two strings
strcmp()	compares two string
strcpy()	copies one string over another
strlen()	finds the length of a string

We shall discuss briefly how each of these functions can be used in the processing of strings.

Copying Strings

We use strcpy() function to copy one string to another. The strcpy function works almost like a string-assignment operator. It takes the form

```
strcpy(string1, string2);
```

and assigns the contents of string2 to string1. string2 may be a character array variable or a string constant. For example, the statement

```
strcpy(city, “DELHI”);
```

will assign the string “DELHI” to the string variable city. Similarly, the statement

```
strcpy(city1, city2);
```

will assign the contents of the string variable ciry2 to the string variable city1. The size of the array city1 should be large enough to receive the contents of city2.

Now can you write a function, which stores your name into a variable called name?

Combining Strings

We can combine two string using strcat function. It takes the following form:

strcat(string1, string2);

String1 and string2 are character arrays. When the function strcat is executed, string2 is appended to string1. it does so by removing the null character at the end of string1 and placing string2 from there. The string at string2 remains unchanged. For example, consider the following three strings:

```
          0 1 2 3 4 5 6 7 8 9 0 1 2
Part1 =   V E  R Y     \0


          0 1 2 3 4 5 6
Part2 =   G OO  D  \0


          0 1 2 3 4 5 6
Part3 =   B A D  \0
```

Execution of the statement

strcat(part1,part2);

will result in:

```
          0 1 2 3 4 5 6 7 8 9 0 1 2
Part1 =   V E  R Y    G O O D  \0


          0 1 2 3 4 5 6
Part2 =   G OO  D  \0
```

While the statement strcat(part1,part3) will result in:

```
          0 1 2 3 4 5 6 7 8 9 0 1 2
Part1 =   V E  R Y    B A D \0
```

	0	1	2	3	4	5	6
Part3 =	B	A	D	\0			

We must make sure that the size of string1 (to which string2 is appended) is large enough to accommodate the final string.

strcat function may also append a string constant to a string variable. The following is valid:

```
strcat(part1, “GOOD”);
```

C permits nesting of strcat functions. For example, the statement

```
strcat(strcat(string1, string2), string3);
```

Is allowed and concatenates all the three strings together. The resultant string is stored in string1.

Now write a C function that joins your surname and name together.

Extracting a portion of strings

This gives a substring from the main string, i.e. copies some of the characters from it. Here is an example for extracting a substring out of a larger string. We might call it like this:

```
#include <stdio.h>
#include <string.h>
main()
{

          char string1[] = "this is a test";
          char string2[10];
          substr(string2, string1, 5, 4);
          printf("%s\n", string2);
}
```

The idea is that we'll extract a substring of length 4, starting at character 5 (0-based) of string1, and copy the substring to string2. Just as with strcpy, it's our responsibility to declare the destination string (string9) big enough. Here is an implementation of substr.

This is an another example which finds whether a string is embedded in other string or not using another string function , strstr.

```
#include <stdio.h>
#include <string.h>

main()
{
  char string[]="string to search";
  char   test[]="sear";

                                    /* strstr returns a pointer into 'string'
                                     * if 'test' is found' if not found, NULL
                                     * is returned.                          */

  if (strstr(string, test)) printf ("String found");

}
```

Comparing two Strings

The strcmp function compares two strings identified by the arguments and has a value() if they are equal. If they are not, if has the numeric difference between the first nonmatching characters in the strings. It takes the form:

```
                strcmp(string, string2);
```

string1 and string2 may be string variables or string constants. Examples are:

```
                strcmp(name1, name2);
```

strcmp(name1, “John”);

strcmp(“Rom”, “Ram”);

Our major concern is to determine whether the strings are equal; if not which is alphabetically above. The value of the mismatch is rarely important. For example the statement

strcmp(“their”, “there”);

Will return a value of –9 which is the numeric difference between ASCII “i” and ASCII “r”. That is, “i” minus “r” in ASCII code is –9. If the value is negative, string1 is alphabetically above string2.

Here is an example to compare two strings.

```
#include <stdio.h>
#include <string.h>

main()
{


char string1[] = "this is";
            char string2[] = "a test";

            if(strcmp(string1, string2) == 0)
                        printf("strings are equal\n");
         else      printf("strings are different\n");

}
```

Now write a C string function that compares the subject taken by you is equal to “CS”?

Finding string length

This function counts and returns the number of characters in a string.

n = strlen(string);

Where n is an integer variable which receives the value of the length of the string. The argument may be a string constant. The counting ends at the first null character.See the following example

```
#include <stdin.h>
#include <string.h>
void main ()
{ string t =  "Rama is the king of Ayodhya”;
  int i;

printf(“%s\n”,t);
  for (i = 0; i<= strlen(t), i++)
    t[i] = toupper(t[i]);
printf(“%s\n”,t);
}
```

The output of the above program is

```
Rama is the king of Ayodhya
RAMA IS THE KING OF AYODHYA
```

toupper() is a function which converts it operand to uppercase letter.
Similarly tolower() is a function which converts it operand to lowercase letter

9.4 Summary

Strings are one-dimensional array of characters terminated by a **null** character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a **null**.

C supports a wide range of functions that manipulate null-terminated string.

Copying string: strcpy(s1, s2); - copies string s2 into string s1

Concatenating string: strcat(s1, s2); - concatenates string s2 onto the end of s1.

String length: strlen(s1); - returns the length of string s1.

String comparison: strcmp(s1, s2); - returns 0 if s1 and s2 are the same; less than 0 if s1<s2; grater than 0 if s1>s2.

String character: strchr(s1, ch); - returns a pointer to the first occurrence of character ch in string s1.

String occurrence: strstr(s1,s2); - returns a pointer to the first occurrence of string s2 in string s1.

9.5 SAQ

10 Marks Questions

1. Write a short note on any two string functions

5Marks Questions

1. Write how two strings can be concatenated.
2. Explain how two strings can be compared.
3. Explain how to extract a portion of a string.

1 Mark Questions

1. The function which helps to add two strings ----- (strcat)
2. The function to compares two strings --- (strcmp)
3. The function to find out the length of the string ---- (strlen)
4. The header file to be included when string functions are used --- (string.h)

Unit – X Functions, Structures and Unions

Structure

10.0 Objective

This unit helps student to clearly understand about what are function, recursions, structures and unions. With the help of this they can write programs easily.

10.1 Introduction

A natural way to solve large problems is to break them down into a series of sub-problems, which can be solved more-or-less independently and then combined to arrive at a complete solution. In programming, this methodology reflects itself in the use of sub-programs or functions Hence C is known as top down modular programming Language.

10.2 Functions

A function is a group of statements that together perform a task. Every Objective-C program has one C function, which is **main()**, and all of the most trivial programs can define additional functions.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is so each function performs a specific task.

A function **declaration** tells the compiler about a function's name, return type, and parameters. A function **definition** provides the actual body of the function.

Basically in Objective-C, we call the function as method. The foundation framework provides numerous built-in methods that your program can call. For example, method **appendString()** to append string to another string.

A method is known with various names like a function or a sub-routine or a procedure, etc.

10.2.1 Function Definition & Usage

C has no procedures, only functions. Their definitions can't be nested but all except main can be called recursively. In C the form of a function definition is

```
<function type><function name>(<formal argument list>)
{
<local variables>

<body>
}
```

See the example below how a function can be written.

```
int mean(int x, int y)
{
int tmp;

tmp = (x + y)/2;
return tmp;
}
```

The above function can be called in main or any other subprograms by referring the function name and passing the parameters.

For example, if you want to call the above procedure for parameters 10 and 15(which are called parameters), you have to call it as follows.

```
mean(10,15);
```

and the 10 is passed to x and 15 is passed to y and the procedure “mean” is executed with the values. Another example shows how to define, write and access a function.

10.2.2 Example Program

Here's a trivial example of a program which includes a user defined function, in this case called "area(...)". The program computes the area of a rectangle of given length and width.

```
        #include<stdio.h>


        int area(int length, int width);                /* function
declaration */
```

```
          /* MAIN PROGRAM: */
          int main()
          {
                    int this_length, this_width;

                    printf( "Enter the length: ");
                    scanf(“%d\n”,  &this_length);
                    printf("Enter the width: ");
                    scanf(“%d\n”,  &this_width);
                     area = area(this_length, this_width);
                    printf( "The area of a  rectangle of length %d and the width
of %d is %d ",  this_length , this_width,, area);

                    return 0;
          }
          /* END OF MAIN PROGRAM */

          /* FUNCTION TO CALCULATE AREA: */
          int area(int length, int width)              /* start of function definition
*/
          {
                    int number;

                    number = length * width;

                    return number;
          }                           /* end of function definition */
          /* END OF FUNCTION */
```

Although this program is not written in the most succinct form possible, it serves to illustrate a number of features concerning functions:

- The structure of a function definition is like the structure of "`main()`", with its own list of variable declarations and program statements.
- A function can have a list of zero or more parameters inside its brackets, each of which can have a separate type.

- A function has to be declared in a function declaration at the top of the program, just after any global constant declarations, and before it can be called by "main()" or in other function definitions.
- Function declarations are a bit like variable declarations - they specify which type the function will return.

A function may have more than one "return" statement, in which case the function definition will end execution as soon as the first "return" is reached. For example:

```
double absolute_value(double number)
{
        if (number >= 0)
                return number;
        else
                return 0 - number;
}
```

10.3Recursions

Recursion is the process of repeating items in a self-similar way. In programming languages, if a program allows you to call a function inside the same function, then it is called a recursive call of the function.

```
Void recursion()

{

Recursion(); /* function calls itself*/

}

Main(0

{

recursion();

}
```

The C programming language supports recursion, i.e., a function to call itself. But while using recursion, programmers need to be careful to define an exit condition from the function, otherwise it will go into an infinite loop.

Recursive functions are very useful to solve many mathematical problems, such as calculating the factorial of a number, generating Fibonacci series, etc.

Now write a program to find Fibonacci series on your own

10.4Structures

Already we have discussed about arrays, which permits us to group elements of the same type into a single logical entity. To reference an element in the array, all that was necessary was that the name of the array b given together with the appropriate subscript. C language provides another tool "Structures" for grouping elements together. Structure is a very powerful concept, which will be used in many C programs.

For example, suppose if we wish to store today's date which is 29/08/2007 inside a program, A natural method for storing the date would be to simply assign the month to an integer variable called month, the day to an integer variable day, and the year to an integer variable year.

```
int month = 08;
int day = 29;
int year = 2007;
```

The above statements would work fine. This is the most acceptable approach. Using this method, we must keep track of three separate variables, which are logically related. It would be much better if we could somehow group these sets of three variables together. In C language, Structures will do this precisely.

10.4.1 Structure Definition

We can define a structure called date, which consists of three components that represent the month, day, and year. The syntax for such a definition is rather straight forward,

Syntax:

```
struct data{
            int month;
            int day;
            int year;
         };
```

The date structure just defined contains three integer members called month, day, and year. The definition of date in a sense defines a new type in the language the variables may subsequently be declared to be of type struct date, as in the declaration.

```
            struct date today;
```

We can also declare a variable purchaseDate to be of the same type by a separate declaration, such as

```
            struct date purchaseDate;
```

or we can simply include two declarations on the same line, as in

```
            struct date today, purchaseDate;
```

Unlike variables of type int, float, or char, a special syntax is needed when dealing with structure variables. A member of a structure is accessed by a specifying the variable name, followed by a period, and then the member name. For example, to set the value of the day in the variable today to 21,

```
today.day = 21;
```

Note that there are no spaces permitted between the variable name, the period, and the member name. To set the year in today to 2007, the expression

```
today.year = 2007;
```

can be used. Let us see how to group time

```
        struct brithtime
        {
                int hour;
                int minutes;
                int seconds;
        };
```

10.4.2 Structure initialization

Initializing structures is similar to initialization of arrays. The elements are simply listed inside a pair of braces with each element separated by a comma. As with arrays most C compilers don't permit automatic structure variables to be initialized, however initialization of global and static structure variables are permitted. A structure variable is made static simply by placing the keyword static directly before the declaration as follows:

```
        static struct date today;
```

To initialize the variable today to August 29, 2007, the statement would be

```
        static struct date today = {8, 29, 2007};
```

can be used. As with the initialization of any static variable, a structure can only be initialized with constant values or constant expressions.

Arrays of Structures:

In C language, arrays not only stores simple data, but also structures. It is perfectly valid to define an array of structures. For example,

```
struct date schedule[10];
```

defines an array called schedule, which consists of 10 elements. Each element inside the array is defined to be of type struct date. Referencing a particular structure element inside the array is quite natural. To set the second schedule inside the schedule array to May 10, 2008, the sequence of statements

```
schedule[1].month =  5;
schedule[1].day = 10;
schedule[1].year = 2008;
```

will works fine.

Structures within Structures

C language provides enormous amount of flexibility in defining structures. For example, it is possible to define a structure that itself contains another structure as one or more of its members, as well as define structures that contain arrays.

Already we have discussed how it is possible to logically group the month, day, and year into a structure called date. The following block of statements is also valid and works fine

```
struct birthdays
{
        struct date bdate;
        struct birthtime btime;
};
```

Variables can now be defined to be of type struct brithdays as follows

```
struct brithdays event;
```

To reference the date structure of the variable even, the syntax is the same:

```
event.bdate;
```

To reference a particular member inside one of these structures, a period followed by the member name must be used as follows

```
event.bdate.month = 8;
```

The even variable can be initialized as

```
Static struct birthdays event = { {8,29,2007}, {5, 10, 35} };
```

This sets the date in the static variable event to August 29, 2007 and sets the time to 5:10:35.

It is also possible to set up the array of type of struct birthdays.

```
Struct birthdays events[50];
```

Structures Containing Arrays:

It is possible to define structures that contain arrays as members. One of the most common applications of this type is in setting up an array of characters inside a structure. For example, suppose we wanted to define a structure called personnel which contains designation and empcode of an employee and further the designation contains 5 characters. The following block of statements will work fine for this job,

```
struct personnel
{
        int empcode;
        char designation[5];
    }
```

This sets up a personnel structure that contains an integer member called empcode and a character member called designation. The member designation actually contains 5 characters. We can define a variable to be of type struct personnel in the normal fashion:

```
    struct personnel finance;
```

we can set the proper fields inside finance for empcode 10 and designation is "MNG-W" with the following sequence of statements:

```
    finance.empcode = 10;
    finance.designation[0]="M"
    finance.designation[1]="N"
    finance.designation[2]="G"
    finance.designation[3]="-"
    finance.designation[4]="W"
```

instead, we can initialize as follows

```
    static struct personnel finance = {10, {"M", "N", "G", "-", "W"}};
```

10.4.3 Example Programs

```
main()
{
    struct date
    {
        int month;
        int day;
        int year
    }
```

```
        struct date today;
        today.month = 8;
        today.day = 29;
        today.year = 2007;

        printf( "To day's date is %d/%d/%d.\n", today.month, today.day,
today.year);
}
```

Output:

```
        Today's date is 8/29/2007.
```

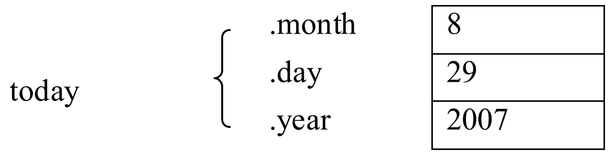
The first statement inside main defines the structure called date to hold three integer member variable called month, day and year. This will simply defines what a date structure looks like to the C compiler, and causes no storage to be reserved inside the computer.

In the second statement, a variable today was declared of type struct date and therefore does cause memory to be reserved for storing the three integer values of the variable today.

Instead of this, we can also define a variable at the same time when the structure is defined. Look at the following statement, which is also correct,

```
struct date
{
        int month;
        int day;
        int year;
} today;
```

After today has been declared, the program then proceeds to assign values to each of these three members of today, as depicted in the following figure:



After the assignments have been made, the values contained inside the structure are displayed by an appropriate printf call.

Structure members follow the same rules as do ordinary variables in the C language. Therefore, any arithmetic operations on members of a structure will be performed as arithmetic operations on an integer variable.

10.5 Unions

Unions are a concept borrowed from structures and therefore follow the same syntax as structures. However, there is major distinction between them in terms of storage. In structures, each member has its own storage location, whereas all the members of a union use the same location. This implies that, although a union may contain many members of different types, it can handle only one member at a time.

10.5.1 Initialization of Unions

Like structures, a union can be declared using the keyword **union** as follows:

```
union item
{
        int m;
        float x;
        char c;
} code;
```

This declares a variable code of type union item. The union contains three members, each with a different data type. However, we can use only one of them at a time. This is due to the fact that only one location is allocated for a union variable, irrespective of its size.

The compiler allocates a piece of storage that is large enough to hold the largest variable type in the union. In the declaration above, the member x requires 4 bytes which is the largest among the members.

10.5.2 Usage of Unions

To access a union member, we can use the same syntax that we use for structure members. That is,

```
code.m
code.x
code.c
```

Are all valid member variables. During accessing, we should make sure that we are accessing the member whose value is currently stored. For example, the statements such as

```
code.m = 379;
code.x = 7859.36;
printf("%d", code.m);
```

Would produce erroneous output (which is machine dependent).

In effect, a union creates a storage location that can be used by any one of its members at a time. When a different member is assigned a new value, the new value supercedes the previous member's value.

Unions may be used in all places where a structure is allowed. The notation for accessing a union member which is nested inside a structure remains the same as for the nested structures.

10.5.3 Example Program

See the program below to use unions.

```
union marks {
  float perc;
  char grade;
}

main () {
  union marks student1;

  student1.perc = 98.5;

  printf( "Marks are %f address is %16lu\n", student1.perc, &student1.perc);


  student1.grade = 'c';

  printf("Grade is %c address is  %16lu\n", student1.grade, &student1.grade);
}
```

Can you change the above program to define a union containing the name of the student, percentage and grade of two students and print the percentage and grade given student's name.

10.7 SAQ

10 marks questions

1. Write a short note on functions.
2. Write a function to add two numbers and call this in the main program to find out the sum of 10 sets of two numbers
3. Write a function to convert a given single number into word and call this in the main function to print the given series of numbers into words.
4. Write a function to multiply two numbers and call this to generate a multiplicative table of a given number.

5. Write a short note on structures.
6. Write a program to accept your date of birth and to print the date of next birth date using structures
7. Write a program to display student's name, mark1,mark2, mark3 and to calculate the average of mark1, mark2 and mark3. (use structures)
8. Write a short note on unions. Explain with examples

5 marks questions

1. What do you know about
   a. Return statement
   b. Parameters
2. Give the general syntax of a function statement
3. Write how you initialize a structure taking an example
4. Write a short note on array of structures

5. Explain how structure and union differ?
6. Explain how unions can be defined and initialized?

1 mark question

1. C uses functions and hence it is known as ------ modular programming language. (Top down)
2. ------ is a self contained block of code that performs particular task. (Function)
3. A function returns a value to the ----- program (Calling)
4. A function can have ---- number of return statements (One or more)
5. A function can have a list of ----- inside its brackets, each of which can have a--- data type. (zero or more parameters, separate)
6. The key word used to declare a structure is ---- (struct)

7. ------ is used o represent data items of single entity ------ (structure)
8. Unions use ---- memory locations. (Common)
9. Key word used to define unions is ------ (union)