DIPLOMA IN COMPUTER APPLICATION (C1244)

NOTES

AICPE®

ALL INDIA COUNCIL FOR PROFESSIONAL EXCELLENCE

DIPLOMA IN COMPUTER APPLICATION (C1244)
(Duration: 1 YEAR)

Course Syllabus

- **Fundamentals of Computer**
- **Introduction to Computer**
  - **History of development of computers**
  - **Computer system concepts**
  - **Characteristics**
  - **Capabilities and limitations**
  - **Generations of computers.**
  - **Basic components of a computer system – Control Unit, ALU, I/ O Devices, memory – RAM, ROM, EPROM, PROM, Flash Memory and other types of memory.**
- **Storage Devices**
  - **Storage fundamentals – Primary Vs Secondary**
  - **Data Storage and Retrieval methods – Sequential, Direct and Index Sequential.**
  - **Various Storage Devices – Magnetic Tape, Magnetic Disks, Cartridge Tape, Data Drives, Hard Disk Drives, Floppy (Winchester Disk), Disks, Optical Disks, CD, VCD, CD-R, CD-RW, Zip Drive, DVD,SVCD.**
- **Computer Software**
  - **Types of Software – System software, Application software, Utility Software, Demoware, Shareware,Freeware, Firmware, Free Software.**
  - **Operating Systems – Functions, Types – Batch Processing, Single User, Multi User, Multiprogramming,Multi-Tasking.**
  - **Programming languages – Machine, Assembly, High Level, 4 GL.**
  - **Data representation in computers.**

    - **Number System of computers – Binary, Octal, Hexa Decimal – Representation & their conversion.**
    - **Coding System – ASCII, BCD, and EBCDIC etc.**
    - **Computer Viruses**
- **Operating System and Application Program**
- **Disk Operating System (DOS)**
    - **DOS basics**
    - **Basic DOS Commands**
- **Windows**
    - **Windows concepts, features, windows structure, desktop, taskbar, start menu, my computer, Recycle Bin.**
    - **Windows Accessories – Calculator, Notepad, Paint, WordPad, Character map.**
    - **Windows Explorer – Creating folders and other Explorer facilities.**
    - **Entertainment – CD Player, DVD Player, Media Player, Sound Recorder, Volume Control.**
- **Linux**
    - **Linux basics**
    - **Basic Linux Commands**
- **Microsoft Office (MS Word, Excel, Power Point)**
    - **Word Processing: MS Word**
    - **Worksheet: MS Excel**
    - **Presentation Graphics: MS Power Point**
- **Programming Language**
- **C Language : Introduction**

  - Introduction
  - C Character Set , Constants, Variables and Key words
- Decision and Control Structure
  - if ,if-elses, forms of if-elses statement
  - Operator
  - Types of Loops
  - Case and Switch
- Arrays and Functions
  - Arrays
  - C Functions – Call by values and Call by reference
- Practical on C Programming
- Each candidate will present a small program with the help of C Programming. Evaluation will be done on the basis of written program, Practical presentation on Computer and Viva.
- Practical on Application Program
- Each candidate will present a small program with the help of Application Program each. Evaluation will be done on the basis of written program, Practical presentation on Computer and Viva.
- Data Base Management System
- .Data Base Management System
- Data Base Tables
- Query Basics and SQL
- Report
- Visual Basics (VB)
- Introduction to Visual Basics Controls
  - VB Interface

    - **Tool Box**
    - **Object Browser Control**
    - **Input Box**
    - **Message Box**
    - **Project Explore**
    - **Standard Control**
    - **Common Dialogue Control**
- **Data Types**
    - **Types of Variables**
    - **Variable declaration**
    - **Array**

VB Function and Sub Routine

    - **Sub routine and functions**
    - **Adding, Removing forms**
    - **Uses of Input Box**
    - **Properties Window**
    - **Debugging**
- **Connectivity with Data Base and Report**
    - **Looping**
    - **Data Controls**
    - **Data Bound Controls**
    - **Data Grids**
    - **Data Reports**
- **Internet and Web Page Designing**

- Internet

    - Evolution, Protocols, Interface Concepts, Internet Vs Intranet, Growth of Internet, ISP, Connectivity – Dial-up, Leased line, VSAT etc. URLs, Domain names, Portals, Application.E-Mail
    - Concepts, POP and WEB Based E-mail, merits, address, Basic of Sending & Receiving, E-mail Protocols, Mailing List, and Free E-mail services, FTP.

- World Wide Web (WWW)

    - History, Working, Web Browsers, Its functions, Concept of Search Engines. Searching the Web, HTTP, URLs, Web Servers, Web Protocols.

- Web Publishing

    - Concepts, Domain name Registration, Space on Host Server for Web site, HTML, Design tools, HTML editors, Image editors, Issues on Web site creations & Maintenance, FTP software for upload web site.

- HTML

    - Concepts of Hypertext, Versions of HTML, Elements of HTML syntax, Head & Body Sections,Building HTML documents, Inserting texts, Images, Hyperlinks, Backgrounds and Colour controls, Different HTML tags, Table layout and presentation, Use of font size & Attributes, List types and its tags.

COMPUTER FUNDAMENTAL :

A computer is an electronic machine that accepts data, stores and processes data into information. The computer is able to work because there are instructions in its memory directing it. The parts of the computer that you can see and touch, such as the keyboard, monitor and the mouse are called hardware. The instructions that direct the computer are called software or computer program.

Data which is raw facts that you the user enter into the computer is called input. This includes; words, numbers, sound and pictures. When the data is entered into the computer, the computer processes the data to produce information which is output. For example, you enter 2+2 into the computer as data, the computer processes it and the result is 4 which is information.

Computers are usually categories into three general categories:

1. Supercomputer – The fastest, largest, most powerful and most expensive computer.
2. Mainframe Computer – This is a little smaller and less powerful than the supercomputer, but, like the supercomputer it is also expensive.
3. Personal Computer (PC)- This is the computer that most people use in their daily lives. This computer is much smaller, less powerful and less expensive than the supercomputer and the mainframe computer. There are two main types of personal computers. Macintosh (Macs) and the PC compatibles (PC) . The main differences between the two are the operating systems and the processor they use. This category of computer has two additional types of computers. These are mobile computer and handheld computer. The most popular type of mobile computer is the notebook or laptop computer, and the handheld computer is a very small PC that you can hold in your hand.

It is important to note that, any computer; regardless of its size has an input device, output device and a system unit.

A BRIEF COMPUTER HISTORY:

The computer as we know it today had its beginning with a 19th century English mathematics professor name Charles Babbage. He designed the Analytical Engine and it was this design that the basic framework of the computers of today are based on.

Generally speaking, computers can be classified into three generations. Each generation lasted for a certain period of time, and each gave us either a new and improved computer or an improvement to the existing computer.

First generation: 1937 – 1946 - In 1937 the first electronic digital computer was built by Dr. John V. Atanas off and Clifford Berry. It was called the Atanas off-Berry Computer (ABC). In 1943 an electronic computer name the Colossus was built for the military. Other developments continued until in 1946 the first general– purpose digital computer, the Electronic Numerical Integrator and Computer (ENIAC) was built. It is said that this computer weighed 30 tons, and had 18,000 vacuum tubes which was used for processing. When this computer was turned on for the first time lights dim in sections of Philadelphia.

Computers of this generation could only perform single task, and they had no operating system.

Second generation: 1947 – 1962 - This generation of computers used transistors instead of vacuum tubes which were more reliable. In 1951 the first computer for commercial use was introduced to the public; the Universal Automatic Computer (UNIVAC 1). In 1953 the International Business Machine (IBM) 650 and 700 series computers made their mark in the computer world. During this generation of computers over 100 computer programming languages were developed, computers had memory and operating systems. Storage media such as tape and disk were in use also were printers for output.

Third generation: 1963 - present - The invention of integrated circuit brought us the third generation of computers. With this invention computers became smaller, more powerful more reliable and they are able to run many different programs at the same time. In1980 Microsoft Disk Operating System (MS-Dos) was born and in 1981 IBM introduced the personal computer (PC) for home and office use. Three years later Apple gave us the Macintosh computer with its icon driven interface and the 90s gave us Windows operating system.

As a result of the various improvements to the development of the computer we have seen the computer being used in all areas of life. It is a very useful tool that will continue to experience new development as time passes.

COMPUTER HARDWARE:

You learned earlier that a computer has electronic and mechanical parts known as hardware. Hardware also includes input devices, output devices, system unit, storage devices and communication devices. Without these components we would not be able to use the computer.

Input Devices - An input device is any hardware component that allows you the user to enter data into the computer.

There are many input devices. Six of the most widely used input devices are:

1. **A keyboard** - You use the keyboard to type letters, numbers, and symbols into the computer.

2. **A Mouse -** The mouse is a pointing device that has a pointer that changes into different shapes as you use the mouse. You click the mouse by pressing and releasing the button. This action allows you to enter data when using a mouse.

3. **A Scanner** -- This input device copies from paper into your computer.

4. **A Microphone** -- The microphone is usually used for voice input into the computer.

5. **A Digital Camera** -- The digital camera allows you to take pictures that you can input into your computer.

6. **A PC Video Camera** -- The PC video camera allows you take both video and still images that you can input onto your computer.

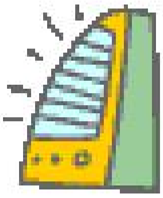
Output Devices - An output device is any hardware component that gives information to the user.

Three commonly used output devices are as follow:

1. **A Monitor --** This output device displays your information on a screen.

2. A Printer -- This output device prints information on paper. This type of printed output is called a hard copy.

2. A Speaker -- Sound is the type of output you will get from a speaker.



The system Unit

The system unit, like the one above is the case that contains all the electronic components of any computer system. The electronic components are considered internal hardware seeing that they are inside the system unit and you cannot see when you look at the computer. These components inside the system unit are what process the data and really makes the computer work. Internal components are as follow:

- **Power Supply:** The power supply converts electricity into the current works for the computer. When the computer is turned on the power supply allows converted electricity to travel to other components inside the computer.
- **Motherboard:** The motherboard is the circuit board that holds the main internal components of the computer together. On the motherboard there are three major cards; sound card that operates the sound, the video card that handles the graphics that you see on the monitor and the modem card which allows computers to communicate with each other.

Also on the motherboard is the Central Processing Unit (CPU), processor or brain of the computer. The CPU controls information and tells the other components inside the computer what to do.

- **RAM & ROM: RAM stands for random access memory**: This memory holds the information you are working with while the computer is turned on. Once you turn the computer off all the information that was in RAM will be gone. ROM stands for read only memory. This memory holds information that you can only read, but not erase. Information in ROM is built in and is always there even when the computer is turned off.
- **Disk Drives:** The disk drive is the device that reads information that is on disk. Generally speaking most computers have three disk drives; hard disk drive, floppy disk drive and CD-Rom drive. However, there computers that have DVD-Rom drive. While the hard disk is hidden inside the computer the floppy and CD-Rom drives are accessible from the front of the system unit.

COMPUTER SOFTWARE:

The computer will not work without software. Software also call programs are the instructions that tell the computer what to do and how o do it. The two main categories of software are system software and application software. The system software also called the operating system (OS) actually runs the computer. This software controls all the operations of the computer and its devices. All computers use system software and without the system software the application software will not work. The most common OS on a PC is the Windows operating system and for the Mac computer it would be the Mac operating system.

Application software is a program that allows users to a specific task on the computer. There are a number of different types of application software available to do many of the tasks we do daily. Four examples of common application software and what they are used for are:

- **Word Processing Application:** One word processing program is Microsoft Word. This program allows you to type letters, assignments and do any other written activity on the computer.
- **Spreadsheet Application:** Microsoft Excel is an example of a spreadsheet program. One can use this program to create charts and do calculations.
- **E-mail Application:** Outlook Express is an e-mail program that allows you to receive and send e-mails.
- **Internet Application:** Internet Explorer is a program that allows you to get connected to the Internet and look at Web sites like the one you are reading now.

It is important to note that when you buy a computer the computer comes with the operating system and some software already installed. You may have to buy more software and install them on the computer. Install means to load the software onto the hard disk of the computer so that you can run or use the software.

Like any other equipment the computer needs to be cared for; let us discuss how we should go about caring for our computer.

COMPUTER CARE:

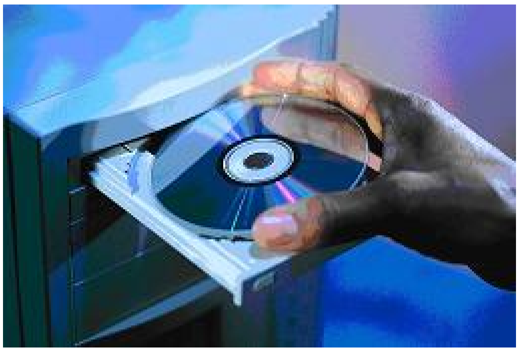
Taking care of your computer is just as important as taking care of your books. Both the internal and the external parts of the computer have to be cared for. Scanning, defragging and reformatting are some of the activities performed to clean up the hard drive. These activities are best left to a grown up and such you should not attempt them. However, there are certain tasks you can perform to ensure you computer is clean; here are a few:

- **Keep Dust Away:** Dust your computer to keep it free of dust and dirt.
- **Keep Food Away:** Do not eat or drink while working on the computer.
- **Use Clean Hands:** Make sure your hands are clean before you type on the keyboard of click the mouse.
- **Treat With Respect:** If you are having problems with your computer, ask for help. Do not bang or hit the computer.

- **Keep Off:** Seeing that the computer is connected to electricity, this means that lightning could be conducted to your computer through the electrical connection. For this reason it is best not to use your computer during a storm.

- **Stop Virus Attack :** A computer virus is a program written by a person on purpose to harm other peoples' computers. A computer virus is passed from one computer o another when you share and download files without the protection of an antivirus software. For this reason you should get permission before downloading files.

- **Handle With Care:** The way you handle your CDs will determine how log they will last. Always hold the CD correctly as shown in the picture below.



INTRODUCTION OF COMPUTER

Computer: A computer is an electronic device, operating under the control of instructions stored in its own memory that can accept data (input), process the data according to specified rules, produce information (output), and store the information for future use .

Functionalities of a computer:

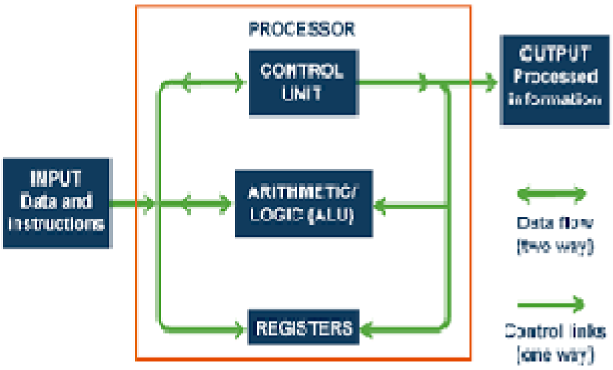
Any digital computer carries out five functions in gross terms:

- Takes data as a Input
- Stores the data / Instructions in its memory and use them when required.
- Processes the data and converts it its into useful information.
- Generates the output.
- Controls all the above four steps.

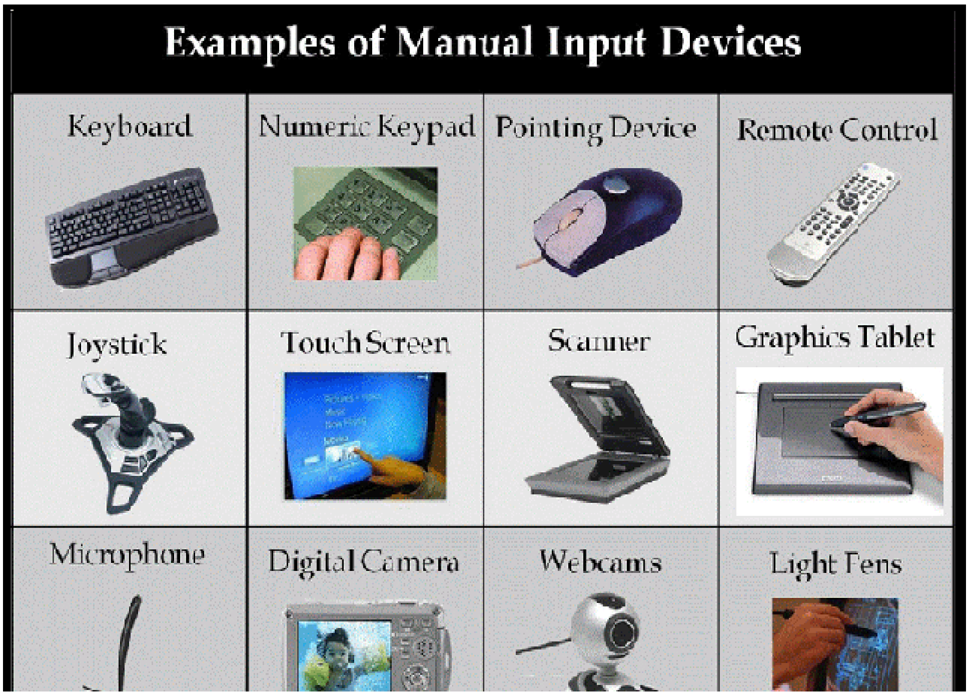
Input
Output
Processing
Memory

Computer Components Any kind of computers consists of **HARDWARE AND SOFTWARE.**

Hardware: Computer hardware is the collection of physical elements that constitutes a computer system. Computer hardware refers to the physical parts or components of a computer such as the monitor, mouse, keyboard, computer data storage, hard drive disk (HDD), system unit (graphic cards, sound cards, memory, motherboard and chips), etc. all of which are physical objects that can be touched.



Input Device: Input device is any peripheral (piece of computer hardware equipment to provide data and control signals to an information processing system such as a computer or other information appliance. Input device Translate data from form that humans understand to one that the computer can work with. Most common are keyboard and mouse.



Example of Input Devices:-

1 . Keyboard	2 . Mouse (pointing device)	3. Microphone
4. Touch screen	5. Scanner	6. Webcam
7. Touchpad	8. MIDI keyboard	9.Graphics Tablets
10.Cameras	11.Pen Input	
12.Video Capture Hardware	13.Microphone	14.Trackballs
15.Barcode reader	16.Digital camera	17.Joystick

18.Gamepad	19.Electronic Whiteboard	

Note: The most common use keyboard is the QWERTY keyboard. Generally standard Keyboard has 104 keys.

Central Processing Unit (CPU)

A CPU is brain of a computer. It is responsible for all functions and processes. Regarding computing power, the CPU is the most important element of a computer system.

The CPU is comprised of three main parts :

* Arithmetic Logic Unit (ALU):

Executes all arithmetic and logical operations. Arithmetic calculations like as addition, subtraction, multiplication and division. Logical operation like compare numbers, letters, or special characters

* Control Unit (CU):

Controls and co-ordinates computer components.

1. Read the code for the next instruction to be executed.

2. Increment the program counter so it points to the next instruction.

3. Read whatever data the instruction requires from cells in memory.

4. Provide the necessary data to an ALU or register.

5. If the instruction requires an ALU or specialized hardware to complete, instruct the hardware to perform the requested operation.

* Registers :

Stores the data that is to be executed next, "very fast storage area".

Primary Memory:-

1. RAM:

Random Access Memory (RAM) is a memory scheme within the computer system responsible for storing data on a temporary basis, so that it can be promptly accessed by the processor as and when needed. It is volatile in nature, which means that data will be erased once supply to the storage device is turned off. RAM stores data randomly and the processor accesses these data randomly from the RAM storage. RAM is considered "random access" because you can access any memory cell directly if you know the row and column that intersect at that cell.

2. ROM:

(Read Only Memory): ROM is a permanent form of storage. ROM stays active regardless of whether power supply to it is turned on or off. ROM devices do not allow data stored on them to be modified.

Secondary Memory:-

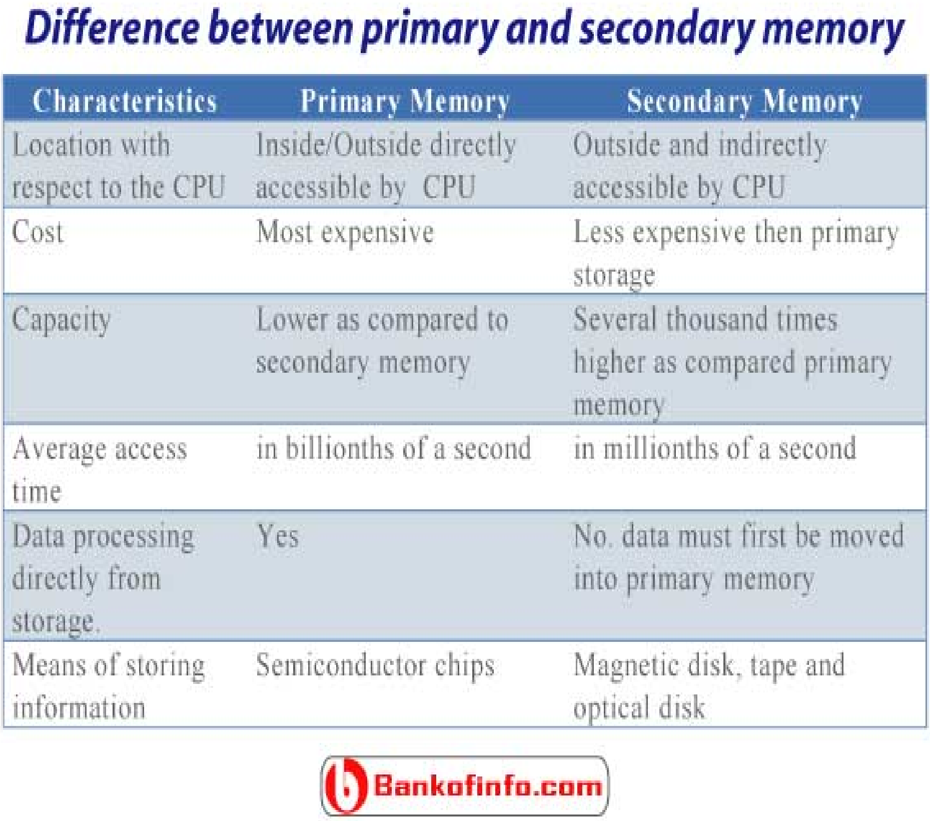
Stores data and programs permanently :its retained after the power is turned off

1. Hard drive (HD): A hard disk is part of a unit, often called a "disk drive," "hard drive," or "hard disk drive," that store and provides relatively quick access to large amounts of data on an electromagnetically charged surface or set of surfaces.

2. **Optical Disk:** an optical disc drive (ODD) is a disk drive that uses laser light as part of the process of reading or writing data to or from optical discs. Some drives can only read from discs, but recent drives are commonly both readers and recorders, also called burners or writers. Compact discs, DVDs, and Blu-ray discs are common types of optical media which can be read and recorded by such drives. Optical drive is the generic name; drives are usually described as "CD" "DVD", or "Bluray", followed by "drive", "writer", etc. There are three main types of optical media: CD, DVD, and Blu-ray disc. CDs can store up to 700 megabytes (MB) of data and DVDs can store up to 8.4 GB of data. Blu-ray discs, which are the newest type of optical media, can store up to 50 GB of data. This storage capacity is a clear advantage over the floppy disk storage media (a magnetic media), which only has a capacity of 1.44 MB.

3. Flash Disk

A storage module made of flash memory chips. A Flash disks have no mechanical platters or access arms, but the term "disk" is used because the data are accessed as if they were on a hard drive. The disk storage structure is emulated.



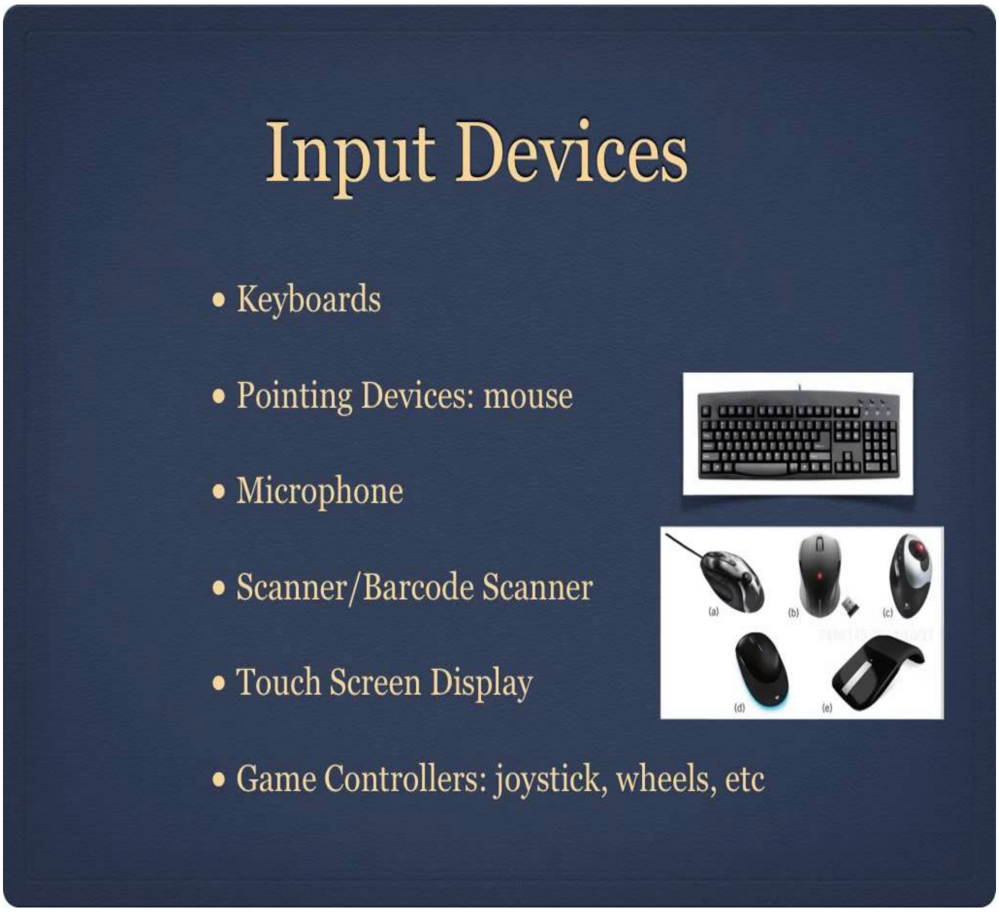
Difference between primary and secondary memory

Characteristics	Primary Memory	Secondary Memory
Location with respect to the CPU	Inside/Outside directly accessible by CPU	Outside and indirectly accessible by CPU
Cost	Most expensive	Less expensive then primary storage
Capacity	Lower as compared to secondary memory	Several thousand times higher as compared primary memory
Average access time	in billionths of a second	in millionths of a second
Data processing directly from storage.	Yes	No. data must first be moved into primary memory
Means of storing information	Semiconductor chips	Magnetic disk, tape and optical disk

Bankofinfo.com

Output devices:

An output device is any piece of computer hardware equipment used to communicate the results of data processing carried out by an information processing system (such as a computer) which converts the electronically generated information into humanreadable form.



Input Devices

- Keyboards
- Pointing Devices: mouse
- Microphone
- Scanner/Barcode Scanner
- Touch Screen Display
- Game Controllers: joystick, wheels, etc

Example of Output Devices:

1. Monitor	2. LCD Projection Panels
3. Printers (all types)	4. Computer Output Microfilm (COM)
5. Plotters	6. Speaker(s)
7. Projector	

Basic types of monitors are :

A. Cathode Ray Tube (CRT).

B. Liquid Crystal Displays (LCD).

C. light - emitting diode (LED).

Printer types:

1 Laser Printer.

2 Ink Jet Printer.

3 Dot Matrix Printer.

Software :

Software is a generic term for organized collections of computer data and instructions, often broken into two major categories: system software that provides the basic non task-specific functions of the computer, and application software which is used by users to accomplish specific tasks.

Software Types :

A. **System software** is responsible for controlling, integrating, and managing the individual hardware components of a computer system so that other software and the users of the system see it as a functional unit without having to be concerned with the low-level details such as transferring data from memory to disk, or rendering text onto a display. Generally, system software consists of an operating system and some fundamental utilities such as disk formatters, file managers, display managers, text editors, user authentication (login) and management tools, and networking and device control software.

B. **Application software** is used to accomplish specific tasks other than just running the computer system. Application software may consist of a single program, such as an image viewer; a small collection of programs (often called a software package) that work closely together to accomplish a task, such as a spreadsheet or text processing system; a larger collection (often called a software suite) of related but independent programs and packages that have a common user interface or shared data format, such as Microsoft Office, which consists of closely integrated word processor, spreadsheet, database, etc.; or a software system, such as a database management system, which is a collection of fundamental programs that may provide some service to a variety of other independent applications.

Unit of Measurements

Storage measurements:

The basic unit used in computer data storage is called a bit (binary digit). Computers use these little bits, which are composed of ones and zeros, to do things and talk to other computers. All your files, for instance, are kept in the computer as binary files and translated into words and pictures by the software (which is also ones and zeros). This two number system, is called a “binary number system” since it has only two numbers in it. The decimal number system in contrast has ten unique digits, zero through nine.

Computer Storage units:

Bit	BIT	0 OR 1
Kilobyte	KB	1024 bytes
Megabyte	MB	1024 kilobytes
Gigabyte	GB	1024 megabytes
Terabyte	TB	1024 gigabytes

Size example

- 1 bit - answer to an yes/no question
- 1 byte - a number from 0 to 255.
- 90 bytes: enough to store a typical line of text from a book.
- 4 KB: about one page of text.
- 120 KB: the text of a typical pocket book.
- 3 MB - a three minute song (128k bit rate)
- 650-900 MB - an CD-ROM
- 1 GB -114 minutes of uncompressed CD-quality audio at 1.4 M bit /s
- 8-16 GB - size of a normal flash drive.

Speed measurement:

The speed of Central Processing Unit (CPU) is measured by Hertz (Hz), Which represent a CPU cycle. The speed of CPU is known as Computer speed.

Computers classification

Computers can be generally classified by size and power as follows, though there is Considerable overlap:

• Personal computer:

A small, single-user computer based on a microprocessor. In addition to the microprocessor, a personal computer has a keyboard for entering data, a monitor for displaying information, and a storage device for saving data.

• Workstation :

A powerful, single-user computer. A workstation is like a personal computer, but it has a more powerful microprocessor and a higher-quality monitor.

• Minicomputer :

A multi-user computer capable of supporting from 10 to hundreds of users simultaneously.

• Mainframe :

A powerful multi-user computer capable of supporting many hundreds or thousands of users simultaneously.

• supercomputer :

An extremely fast computer that can perform hundreds of millions of instructions per second.

Laptop and Smartphone Computers

LAPTOP:

A laptop is a battery or AC-powered personal computer that can be easily carried and used in a variety of locations. Many laptops are designed to have all of the functionality of a desktop computer, whichmeans they can generally run the same software and open the same types of files. However, some laptops, such as netbooks, sacrifice some functionality in order to be even more portable.

Notebook:

A notebook is a type of laptop that is designed to be even more portable. Notebooks are often cheaper than laptops or desktops. They are generally less powerful than other types of computers, but they provide enough power for email and internet access, which is where the name "notebook" comes from.

Mobile Device:

A mobile device is basically any handheld computer. It is designed to be extremely portable, often fitting in the palm of your hand or in your pocket. Some mobile devices are more powerful, and they allow you to do many of the same things you can do with a desktop or laptop computer. These include tablet computers, e-readers, and smartphone.

Tablet Computers:

Like laptops, tablet computers are designed to be portable. However, they provide a very different computing experience. The most obvious difference is that tablet computers don't have keyboards or touchpads. Instead, the entire screen is touch-sensitive, allowing you to type on a virtual keyboard and use your finger as a mouse pointer. Tablet computers are mostly designed for consuming media, and they are optimized for tasks like web browsing, watching videos, reading e-books, and playing games. For many people, a "regular" computer like a desktop or laptop is still needed in order to use some programs. However, the convenience of a tablet computer means that it may be ideal as a second computer.

Smartphones:

A smartphone is a powerful mobile phone that is designed to run a variety of applications in addition to phone service. They are basically small tablet computers, and they can be used for web browsing, watching videos, reading e-books, playing games and more.

Data, Information and Knowledge Data:

Facts and figures which relay something specific, but which are not organized in any way and which provide no further information regarding patterns, context, etc. So data means "unstructured facts and figures that have the least impact on the typical manager."

Information:

For data to become information, it must be contextualized, categorized, calculated and condensed. Information thus paints a bigger picture; it is data with relevance and purpose. It may convey a trend in the environment, or perhaps indicate a pattern of sales for a given period of time. Essentially information is found "in answers to questions that begin with such words as who, what, where, when, and how many".

Knowledge:

Knowledge is closely linked to doing and implies know-how and understanding. The knowledge possessed by each individual is a product of his experience, and encompasses the norms by which he evaluates new inputs from his surroundings.

The content of the human mind can be classified into four categories:

1. **Data**: symbols
2. **Information**: data that are processed to be useful; provides answers to "who", "what", "where", and "when" questions
3. **Knowledge:** application of data and information; answers "how" questions
4. **Wisdom:** evaluated understanding.

We need to understand that processing data produced Information and process Information produces Knowledge and so on .

Characteristics of Computer Speed, accuracy, diligence, storage capability and versatility are some of the key characteristics of a computer. A brief overview of these characteristics are :

• **Speed:** The computer can process data very fast, at the rate of millions of instructions per second. Some calculations that would have taken hours and days to complete otherwise, can be completed in a few seconds using the computer. For example, calculation and generation of salary slips of thousands of employees of organization, weather forecasting that requires analysis of a large amount of data related to temperature, pressure and humidity of various places, etc.

• Accuracy:

Computer provides a high degree of accuracy. For example, the computer can accurately give the result of division of any two numbers up to 10 decimal places.

• Diligence:

When used for a longer period of time, the computer does not get tired or fatigued. It can perform long and complex calculations with the same speed and accuracy from the start till the end.

• Storage Capability:

Large volumes of data and information can be stored in the computer and also retrieved whenever required. A limited amount of data can be stored, temporarily, in the primary memory. Secondary storage devices like floppy disk and compact disk can store a large amount of data permanently.

• Versatility:

Computer is versatile in nature. It can perform different types of tasks with the same ease. At one moment you can use the computer to prepare a letter document and in the next moment you may play music or print a document. Computers have several limitations too. Computer can only perform tasks that it has been programmed to do. 10 Computer cannot do any work without instructions from the user. It executes instructions as specified by the user and does not take its own decisions.

Computer Viruses

- **Viruses**:

  A virus is a small piece of software that piggybacks on real programs. For example, a virus might attach itself to a program such as a spreadsheet program. Each time the spreadsheet program runs, the virus runs, too, and it has the chance to reproduce (by attaching to other programs) or wreak havoc.

- **E-mail viruses:**

  An e-mail virus travels as an attachment to e-mail messages, and usually replicates itself by automatically mailing itself to dozens of people in the victim's e-mail address book. Some e-mail viruses don't even require a double-click -- they launch when you view the infected message in the preview pane of your e-mail software [source: Johnson].

- **Trojan horses:**

  A Trojan horse is simply a computer program. The program claims to do one thing (it may claim to be a game) but instead does damage when you run it (it may erase your hard disk). Trojan horses have no way to replicate automatically.

- **Worms:**

  A worm is a small piece of software that uses computer networks and security holes to replicate itself. A copy of the worm scans the network for another machine that has a specific security hole. It copies itself to the new machine using the security hole, and then starts replicating from there, as well.

What are some tips to avoid viruses and lessen their impact:

- Install anti-virus software from a reputable vendor. Update it and use it regularly.
- In addition to scanning for viruses on a regular basis, install an "on access" scanner (included in most anti-virus software packages) and configure it to start each time you start up your computer. This will protect your system by checking for viruses each time you run an executable file.
- Use a virus scan before you open any new programs or files that may contain executable code. This includes packaged software that you buy from the store as well as any program you might download from the Internet.
- If you are a member of an online community or chat room, be very careful about accepting files or clicking links that you find or that people send you within the community.
- Make sure you back up your data (documents, bookmark files, important email messages, etc.) on disc so that in the event of a virus infection, you do not lose valuable work.

Computer capabilities and limitations

❖ Like all machines, a computer needs to be directed and controlled in order to perform a task successfully. Until such time as a program is prepared and stored in the computer's memory, the computer 'knows' absolutely nothing, not even how to accept or reject data. Even the most sophisticated computer, no matter how capable it is, must be told what to do. Until the capabilities and the limitations of a computer are recognized, its usefulness cannot be thoroughly understood.

❖ In the first place, it should be recognized that computers are capable of doing repetitive operations. A computer can perform similar operations thousands of times, without becoming bored, tired, or even careless.

❖ Secondly, computers can process information at extremely rapid rates. For example, modern computers can solve certain classes of arithmetic problems millions of times faster than a skilled mathematician. Speeds for performing decision-making operations are comparable to those for arithmetic operations but input-output operations, however, involve mechanical motion and hence require more time. On a typical computer system, cards are read at an average speed of 1000 cards per minute and as many as 1000 lines can be printed at the same rate.

- Thirdly, computers may be programmed to calculate answers to whatever level of accuracy is specified by the programmer. In spite of newspaper headlines such as ‘Computer Fails’, these machines are very accurate and reliable especially when the number of operations they can perform every second is considered. Because they are man-made machines, they sometimes malfunction or break down and have to be repaired. However, in most instances when the computer fails, it is due to human error and is not the fault of the computer at all.

- In the fourth place, general-purpose computers can be programmed to solve various types of problems because of their flexibility. One of the most important reasons why computers are so widely use today is that almost every big problem can be solved by solving a number of little problems—one after another.

- Finally, a computer, unlike a human being, has no intuition. A person may suddenly find the answer to a problem without working out too many of the details, but a computer can only proceed as it has been programmed to.

- Using the very limited capabilities possessed by all computers, the task of producing a university payroll, for instance, can be done quite easily. The following kinds of things need to be done for each employee on the payroll. First: Input information about the employee such as wage rate, hours worked, tax rate, unemployment insurance, and pension deductions. Second: Do some simple arithmetic and decision making operations. Third: Output a few printed lines on a cheque. By repeating this process over and over again, the payroll will eventually be completed.

System Components

A modern PC is both simple and complicated. It is simple in the sense that over the years, many of the components used to construct a system have become integrated with other components into fewer and fewer actual parts. It is complicated in the sense that each part in a modern system performs many more functions than did the same types of parts in older systems.

This section briefly examines all the components and peripherals in a modern PC system. Each item is discussed further in later chapters.

Here are the components and peripherals necessary to assemble a basic modern PC system:

- Motherboard
- Processor
- Memory (RAM)
- Case/chassis
- Power supply
- Floppy drive
- Hard disk
- CD-ROM, CD-RW, or DVD-ROM drive
- Keyboard
- Mouse
- Video card
- Monitor (display)
- Sound card
- Speakers
- Modem

Component	**Description**
Motherboard	The motherboard is the core of the system. It really is the PC; everything else is connected to it, and it controls everything in the system, "Microprocessor Types and Specifications."
Processor	The processor is often thought of as the "engine" of the computer. It's also called the CPU (central processing unit).
Memory (RAM)	The system memory is often called RAM (for random access memory). This is the primary memory, which holds all the programs and data the processor is using at a given time.
Case/chassis	The case is the frame or chassis that houses the motherboard, power supply, disk drives, adapter cards, and any other physical components in the system. "Power Supply and Chassis/Case."
Power supply	The power supply is what feeds electrical power to every single part in the PC.
Floppy drive	The floppy drive is a simple, inexpensive, low-capacity, removable-media, magnetic storage device.
Hard drive	The hard disk is the primary archival storage memory for the system.
CD-ROM/DVD-ROM	CD-ROM (compact disc read-only) and DVD-ROM (digital versatile disc read-only) drives are relatively high-capacity, removable media, optical drives

Keyboard	The keyboard is the primary device on a PC that is used by a human to communicate with and control a system.
Mouse	Although many types of pointing devices are on the market today, the first and most popular device for this purpose is the mouse.

Computer Memory :

Memory is an internal storage area in a computer, which is availed to store data and programs either permanently or temporarily. Computer memory is broadly divided into two groups and they are:

- Primary memory and
- Secondary memory

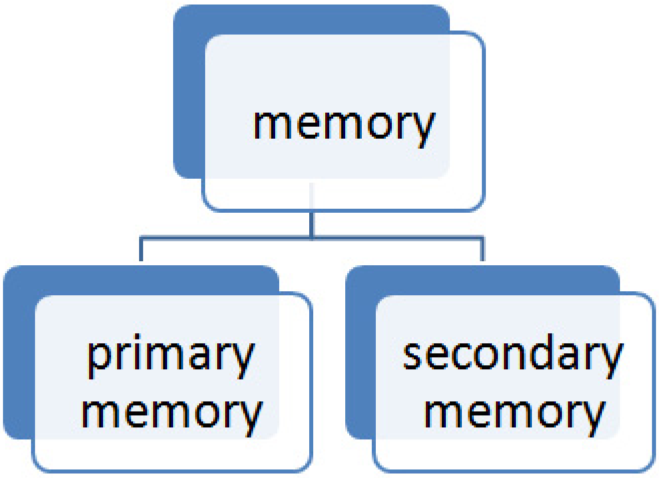


Fig1: Classification of the Computer Memory

When the main memory holds instructions and data when a program is executing, the auxiliary memory or secondary memory holds data and programs which are not currently in use and furnishes long term storage.

The primary memory and secondary memory are further classified into distinct groups and those are explained in the below diagram:

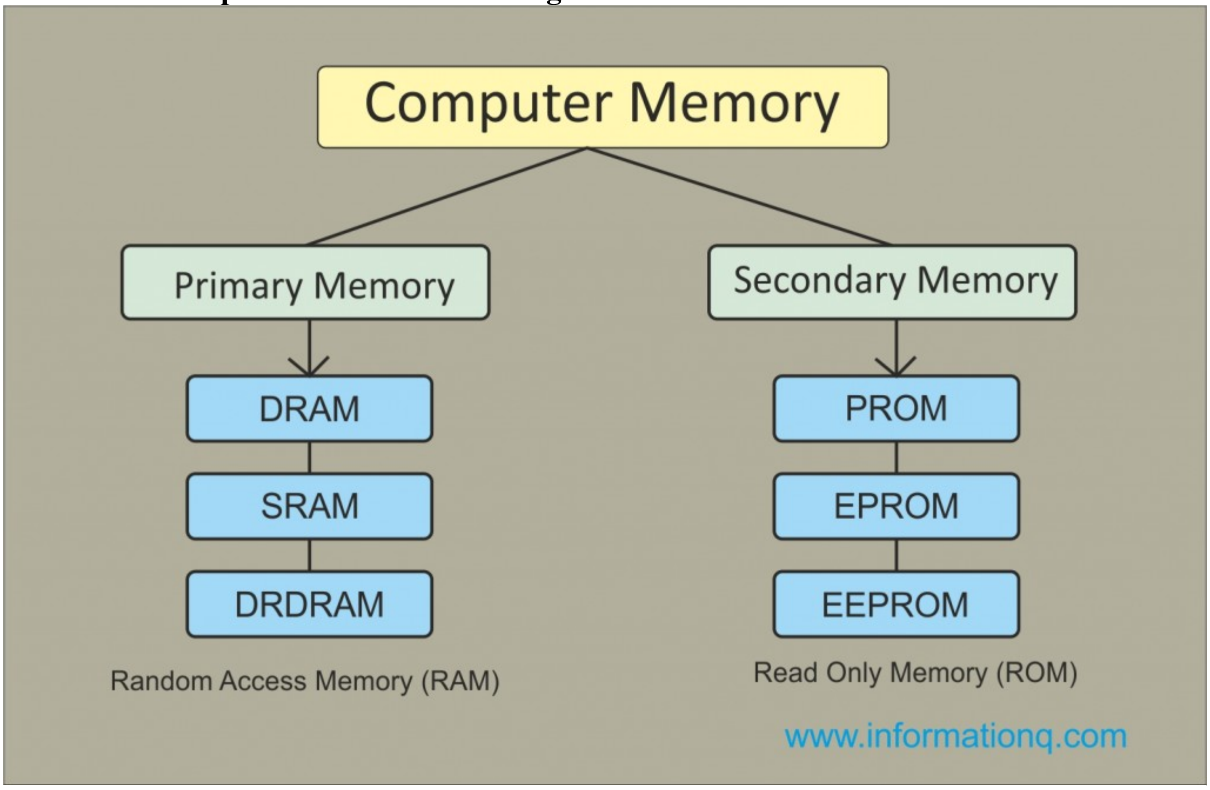


Fig2: Classification of Primary Memory and Secondary Memory

Primary memory:

Primary memory is the only type of memory which is directly accessed by the CPU. The CPU continuously reads instructions stored in the primary memory and executes them. Any data that has to be operated by the CPU is also stored. The information is transferred to various locations through the BUS.

Primary memories are of two types. They are:

- RAM
- ROM

1. **RAM:**
   It stands for Random Access Memory. Here data can be stored temporarily, so this type of memory is called as temporary memory or volatile memory because when power fails the data from RAM will be erased. The information stored in the RAM is basically loaded from the computer's disk and includes information related to the operating system and applications that are currently executed by the processor. RAM is considered random access because any memory cell can be directly accessed if its address is known. RAM is of distinct types like SRAM, DRAM, and VRAM.

2. **ROM:**
   It stands for Read Only Memory. In this, the data will be furnished by the manufacturers regarding the system, so this information can simply be read by the user but cannot add new data or it cannot be modified. ROMs are of distinct types:

- **PROM** – Programmable Read Only Memory

- **EPROM** – Erasable Programmable Read Only Memory

- **EEPROM** – Electrically Erasable Programmable Read Only Memory

Secondary memory:

Secondary memory or auxiliary memory consists of slower and less expensive device that communicates indirectly with CPU via main memory. The secondary memory stores the data and keeps it even when the power fails. It is used to store or save large data or programs or other information.

The secondary storage devices are explained below:

- **Magnetic disks**
- **Magnetic tape**
- **Optical disk**
- **USB flash drive**
- **Mass storage devices**

1.Magnetic Disks:

Magnetic disks are made of rigid metals or synthetic plastic material. The disk platter is coated on both the surfaces with magnetic material and both the surfaces can be used for storage. The magnetic disk furnishes direct access and is for both small and large computer systems. The magnetic disk comes in two forms:

- Floppy disks
- Hard disks

2.Magnetic Tape:

Magnetic tape is serial access storage medium and it can store a large volume of data at low costs. The conventional magnetic tape is in reels of up to 3600 feet made of Mylar plastic tape. The tape is one-half inch in width and is coated with magnetic material on one side. The reel of tape is loaded on a magnetic tape drive unit. During any read/write operation, the tape is moved from one spool to another in the same way as in the audiocassette tape recorder. The magnetic tape is densely packed with magnetic spots in frames across its width.

3.Optical Drives:

optical drives 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 6GB. Optical store devices are the most widely used and reliable storage devices. The most widely used type of optical storage devices are explained below:

- **CD – ROM**
- **DVD – ROM**
- **CD – RECORDABLE**
- **CD – REWRITABLE**
- **PHOTO – CD**

4.USB flash Drives:

USB flash drives are removable, rewritable and are physically much smaller drives, which have the weight of less than 30g. In the year of 2010, the storage capacity of the USB flash drives was as large as 256GB. Such devices are a good substitute for floppy disks and CD – ROMs as they are smaller, faster, have thousands of times more capacity, and are more durable and reliable. Until 2005, most desktop and laptop computers had floppy disk drives, but nowadays floppy disk drives have been abandoned in favor of USB ports. The USB connector is often protected inside a removable cap, although it is not likely to be damaged if unprotected. USB flash drives draw power from the computer through external USB connection. The most widely used USB flash drives are the memory cards.

5.Mass Storage Devices:

Mass storage devices refer to the saving of huge data in a persistent manner. Mass storage machines can store up to several trillion bytes of data and hence are used to store or save large databases, such as the information of customers of a big retail chain and library transactions of students in a college. Some of the commonly used mass storage devices are explained below:

- **Disk array**
- **Automated tape**
- **CD – ROM jukebox**

Storage Devices:

Storage fundamentals – Primary Vs Secondary:

Computer data storage, often called storage or memory, is a technology consisting of computer components and recording media that are used to retain digital data. It is a core function and fundamental component of computers.

The central processing unit (CPU) of a computer is what manipulates data by performing computations. In practice, almost all computers use a storage hierarchy:468–473 which puts fast but expensive and small storage options close to the CPU and slower but larger and cheaper options farther away. Generally the fast volatile technologies (which lose data when off power) are referred to as "memory", while slower persistent technologies are referred to as "storage".

In the Von Neumann architecture, the CPU consists of two main parts: The control unit and the arithmetic logic unit (ALU). The former controls the flow of data between the CPU and memory, while the latter performs arithmetic and logical operations on data.

Hierarchy of storage:

Generally, the lower a storage is in the hierarchy, the lesser its bandwidth and the greater its access latency is from the CPU. This traditional division of storage to primary, secondary, tertiary and off-line storage is also guided by cost per bit.

In contemporary usage, "memory" is usually semiconductor storage read-write random-access memory, typically DRAM (dynamic RAM) or other forms of fast but temporary storage. "Storage" consists of storage devices and their media not directly accessible by the CPU (secondary or tertiary storage), typically hard disk drives, optical disc drives, and other devices slower than RAM but non-volatile (retaining contents when powered down).

Historically, memory has been called core memory, main memory, real storage or internal memory. Meanwhile, non-volatile storage devices have been referred to as secondary storage, external memory or auxiliary/peripheral storage.

Primary storage:

Primary storage (also known as main memory or internal memory), often referred to simply as memory, is the only one directly accessible to the CPU. The CPU continuously reads instructions stored there and executes them as required. Any data actively operated on is also stored there in uniform manner. Historically, early computers used delay lines, Williams tubes, or rotating magnetic drums as primary storage. By 1954, those unreliable methods were mostly replaced by magnetic core memory. Core memory remained dominant until the 1970s, when advances in integrated circuit technology allowed semiconductor memory to become economically competitive. This led to modern random-access memory (RAM). It is small-sized, light, but quite expensive at the same time. (The particular types of RAM used for primary storage are also volatile, i.e. they lose the information when not powered).As shown in the diagram, traditionally there are two more sub-layers of the primary storage, besides main large-capacity RAM.

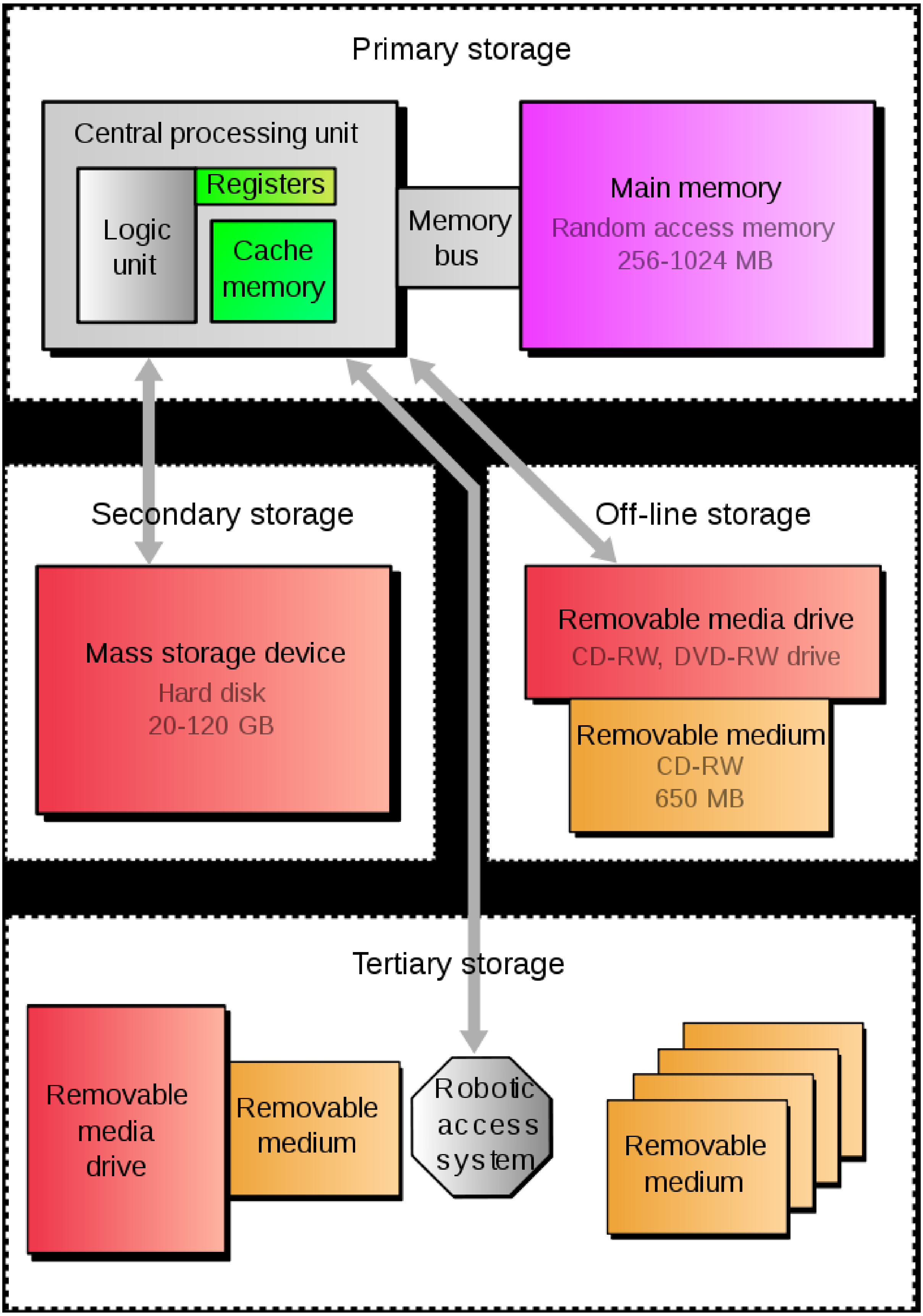
Processor registers are located inside the processor. Each register typically holds a word of data (often 32 or 64 bits). CPU instructions instruct the arithmetic logic unit to perform various calculations or other operations on this data (or with the help of it). Registers are the fastest of all forms of computer data storage.

Processor cache is an intermediate stage between ultra-fast registers and much slower main memory. It was introduced solely to improve the performance of computers. Most actively used information in the main memory is just duplicated in the cache memory, which is faster, but of much lesser capacity. On the other hand, main memory is much slower, but has a much greater storage capacity than processor registers. Multi-level hierarchical cache setup is also commonly used—primary cache being smallest, fastest and located inside the processor; secondary cache being somewhat larger and slower.

Main memory is directly or indirectly connected to the central processing unit via a memory bus. It is actually two buses (not on the diagram): an address bus and a data bus. The CPU firstly sends a number through an address bus, a number called memory address, that indicates the desired location of data. Then it reads or writes the data in the memory cells using the data bus. Additionally, a memory management unit (MMU) is a small device between CPU and RAM recalculating the actual memory address, for example to provide an abstraction of virtual memory or other tasks.

As the RAM types used for primary storage are volatile (uninitialized at start up), a computer containing only such storage would not have a source to read instructions from, in order to start the computer. Hence, non-volatile primary storage containing a small start up program (BIOS) is used to bootstrap the computer, that is, to read a larger program from non-volatile secondary storage to RAM and start to execute it. A non-volatile technology used for this purpose is called ROM, for read-only memory (the terminology may be somewhat confusing as most ROM types are also capable of random access).

Many types of "ROM" are not literally read only, as updates to them are possible; however it is slow and memory must be erased in large portions before it can be re-written. Some embedded systems run programs directly from ROM (or similar), because such programs are rarely changed. Standard computers do not store non-rudimentary programs in ROM, and rather, use large capacities of secondary storage, which is non-volatile as well, and not as costly. Recently, primary storage and secondary storage in some uses refer to what was historically called, respectively, secondary storage and tertiary storage



Secondary storage:

Secondary storage (also known as external memory or auxiliary storage), differs from primary storage in that it is not directly accessible by the CPU. The computer usually uses its input/output channels to access secondary storage and transfers the desired data using intermediate area in primary storage. Secondary storage does not lose the data when the device is powered down—it is non-volatile. Per unit, it is typically also two orders of magnitude less expensive than primary storage. Modern computer systems typically have two orders of magnitude more secondary storage than primary storage and data are kept for a longer time there.

In modern computers, hard disk drives are usually used as secondary storage. The time taken to access a given byte of information stored on a hard disk is typically a few thousandths of a second, or milliseconds. By contrast, the time taken to access a given byte of information stored in random-access memory is measured in billionths of a second, or nanoseconds. This illustrates the significant access-time difference which distinguishes solid-state memory from rotating magnetic storage devices: hard disks are typically about a million times slower than memory. Rotating optical storage devices, such as CD and DVD drives, have even longer access times. With disk drives, once the disk read/write head reaches the proper placement and the data of interest rotates under it, subsequent data on the track are very fast to access. To reduce the seek time and rotational latency, data are transferred to and from disks in large contiguous blocks.

When data reside on disk, accessing them in large blocks to hide latency offers an opportunity to design efficient external memory algorithms. Sequential or block access on disks is orders of magnitude faster than random access, and many sophisticated paradigms have been developed to design efficient algorithms based upon sequential and block access. Another way to reduce the I/O bottleneck is to use multiple disks in parallel in order to increase the bandwidth between primary and secondary memory.

Some other examples of secondary storage technologies are flash memory (e.g. USB flash drives or keys), floppy disks, magnetic tape, paper tape, punched cards, standalone RAM disks, and Iomega Zip drives.

The secondary storage is often formatted according to a file system format, which provides the abstraction necessary to organize data into files and directories, providing also additional information (called metadata) describing the owner of a certain file, the access time, the access permissions, and other information.

Most computer operating systems use the concept of virtual memory, allowing utilization of more primary storage capacity than is physically available in the system. As the primary memory fills up, the system moves the least-used chunks (pages) to secondary storage devices (to a swap file or page file), retrieving them later when they are needed. As more of these retrievals from slower secondary storage are necessary, the more the overall system performance is degraded.

A hard disk drive with protective cover removed.

Tertiary storage:

Tertiary storage or tertiary memory provides a third level of storage. Typically, it involves a robotic mechanism which will mount (insert) and dismount removable mass storage media into a storage device according to the system's demands; such data are often copied to secondary storage before use. It is primarily used for archiving rarely accessed information since it is much slower than secondary storage . This is primarily useful for extraordinarily large data stores, accessed without human operators. Typical examples include tape libraries and optical jukeboxes.

When a computer needs to read information from the tertiary storage, it will first consult a catalog database to determine which tape or disc contains the information. Next, the computer will instruct a robotic arm to fetch the medium and place it in a drive. When the computer has finished reading the information, the robotic arm will return the medium to its place in the library.

Tertiary storage is also known as near line storage because it is "near to online". The formal distinction between online, near line, and offline storage is Online storage is immediately available for I/O. Near line storage is not immediately available, but can be made online quickly without human intervention. Offline storage is not immediately available, and requires some human intervention to become online.

For example, always-on spinning hard disk drives are online storage, while spinning drives that spin down automatically, such as in massive arrays of idle disks (MAID), are near line storage. Removable media such as tape cartridges that can be automatically loaded, as in tape libraries, are near line storage, while tape cartridges that must be manually loaded are offline storage.

A large tape library, with tape cartridges placed on shelves in the front, and a robotic arm moving in the back. Visible height of the library is about 180 cm.

Off-line storage:

Off-line storage is a computer data storage on a medium or a device that is not under the control of a processing unit. The medium is recorded, usually in a secondary or tertiary storage device, and then physically removed or disconnected. It must be inserted or connected by a human operator before a computer can access it again. Unlike tertiary storage, it cannot be accessed without human interaction.

Off-line storage is used to transfer information, since the detached medium can be easily physically transported. Additionally, in case a disaster, for example a fire, destroys the original data, a medium in a remote location will probably be unaffected, enabling disaster recovery. Off-line storage increases general information security, since it is physically inaccessible from a computer, and data confidentiality or integrity cannot be affected by computer-based attack techniques. Also, if the information stored for archival purposes is rarely accessed, off-line storage is less expensive than tertiary storage.

In modern personal computers, most secondary and tertiary storage media are also used for off-line storage. Optical discs and flash memory devices are most popular, and to much lesser extent removable hard disk drives. In enterprise uses, magnetic tape is predominant. Older examples are floppy disks, Zip disks, or punched cards.

Characteristics of Storage:

Storage technologies at all levels of the storage hierarchy can be differentiated by evaluating certain core characteristics as well as measuring characteristics specific to a particular implementation. These core characteristics are volatility, mutability, accessibility, and addressability. For any particular implementation of any storage technology, the characteristics worth measuring are capacity and performance.

Volatility:

Non-volatile memory retains the stored information even if not constantly supplied with electric power.[8] It is suitable for long-term storage of information. Volatile memory requires constant power to maintain the stored information. The fastest memory technologies are volatile ones, although that is not a universal rule. Since the primary storage is required to be very fast, it predominantly uses volatile memory.

Dynamic random-access memory is a form of volatile memory that also requires the stored information to be periodically reread and rewritten, or refreshed, otherwise it would vanish. Static random-access memory is a form of volatile memory similar to DRAM with the exception that it never needs to be refreshed as long as power is applied; it loses its content when the power supply is lost.

An uninterruptible power supply (UPS) can be used to give a computer a brief window of time to move information from primary volatile storage into non-volatile storage before the batteries are exhausted. Some systems, for example EMC Symmetrix, have integrated batteries that maintain volatile storage for several minutes.

Mutability:

Read/write storage or mutable storage : Allows information to be overwritten at any time. A computer without some amount of read/write storage for primary storage purposes would be useless for many tasks. Modern computers typically use read/write storage also for secondary storage.

Read only storage : Retains the information stored at the time of manufacture, and write once storage (write once read many) allows the information to be written only once at some point after manufacture. These are called immutable storage. Immutable storage is used for tertiary and off-line storage. Examples include CD-ROM and CD-R.

Slow write, fast read storage : Read/write storage which allows information to be overwritten multiple times, but with the write operation being much slower than the read operation. Examples include CD-RW and swayne memory.

Accessibility:

Random Access:

Any location in storage can be accessed at any moment in approximately the same amount of time. Such characteristic is well suited for primary and secondary storage. Most semiconductor memories and disk drives provide random access.

Sequential Access:

The accessing of pieces of information will be in a serial order, one after the other; therefore the time to access a particular piece of information depends upon which piece of information was last accessed. Such characteristic is typical of off-line storage.

Addressability:

Location-addressable :

Each individually accessible unit of information in storage is selected with its numerical memory address. In modern computers, location-addressable storage usually limits to primary storage, accessed internally by computer programs, since location-addressability is very efficient, but burdensome for humans.

File addressable:

Information is divided into files of variable length, and a particular file is selected with human-readable directory and file names. The underlying device is still location-addressable, but the operating system of a computer provides the file system abstraction to make the operation more understandable. In modern computers, secondary, tertiary and off-line storage use file systems.

Content-addressable:

Each individually accessible unit of information is selected based on the basis of (part of) the contents stored there. Content-addressable storage can be implemented using software (computer program) or hardware (computer device), with hardware being faster but more expensive option. Hardware content addressable memory is often used in a computer's CPU cache.

Capacity:

Raw capacity :

The total amount of stored information that a storage device or medium can hold. It is expressed as a quantity of bits or bytes (e.g. 10.4 megabytes).

Memory storage density:

The compactness of stored information. It is the storage capacity of a medium divided with a unit of length, area or volume (e.g. 1.2 megabytes per square inch).

Performance:

Latency:

The time it takes to access a particular location in storage. The relevant unit of measurement is typically nanosecond for primary storage, millisecond for secondary storage, and second for tertiary storage. It may make sense to separate read latency and write latency (especially for non-volatile memory[8]) and in case of sequential access storage, minimum, maximum and average latency**.**

Throughput:

The rate at which information can be read from or written to the storage. In computer data storage, throughput is usually expressed in terms of megabytes per second (MB/s), though bit rate may also be used. As with latency, read rate and write rate may need to be differentiated. Also accessing media sequentially, as opposed to randomly, typically yields maximum throughput**.**

Granularity:

The size of the largest "chunk" of data that can be efficiently accessed as a single unit, e.g. without introducing additional latency.

Reliability:

The probability of spontaneous bit value change under various conditions, or overall failure rate.Utilities such as hdparm and sar can be used to measure IO performace in Linux.

Energy use:

Storage devices that reduce fan usage, automatically shut-down during inactivity, and low power hard drives can reduce energy consumption by 90 percent.

2.5-inch hard disk drives often consume less power than larger ones.Low capacity solid-state drives have no moving parts and consume less power than hard disks.Also, memory may use more power than hard disks. Large caches, which are used to avoid hitting memory wall, may also consume a large amount of power.

Magnetic storage Media:

uses different patterns of magnetization on a magnetically coated surface to store information. Magnetic storage is non-volatile. The information is accessed using one or more read/write heads which may contain one or more recording transducers. A read/write head only covers a part of the surface so that the head or medium or both must be moved relative to another in order to access data. In modern computers, magnetic storage will take these forms:

Magnetic disk:

- Floppy disk, used for off-line storage
- Hard disk drive, used for secondary storage
- Magnetic tape, used for tertiary and off-line storage
- Carousel memory (magnetic rolls)

In early computers, magnetic storage was also used as:

- Primary storage in a form of magnetic memory, or core memory, core rope memory, thin-film memory and/or twistor memory.
- Tertiary (e.g. NCR CRAM) or off line storage in the form of magnetic cards.
- Magnetic tape was then often used for secondary storage.

Optic Storage media: The typical optical disc, stores information in deformities on the surface of a circular disc and reads this information by illuminating the surface with a laser diode and observing the reflection. Optical disc storage is non-volatile. The deformities may be permanent (read only media),formed once (write once media) or reversible (recordable or read/write media). The following forms are currently in common use:[21]

- CD, CD-ROM, DVD, BD-ROM: Read only storage, used for mass distribution of digital information (music, video, computer programs)
- CD-R, DVD-R, DVD+R, BD-R: Write once storage, used for tertiary and off-line storage
- CD-RW, DVD-RW, DVD+RW, DVD-RAM, BD-RE: Slow write, fast read storage, used for tertiary and off-line storage
- Ultra Density Optical or UDO is similar in capacity to BD-R or BD-RE and is slow write, fast read storage used for tertiary and off-line storage. Magneto-optical disc storage is optical disc storage where the magnetic state on a ferromagnetic surface stores information. The information is read optically and written by combining magnetic and optical methods. Magneto-optical disc storage is non-volatile, sequential access, slow write, fast read storage used for tertiary and off-line storage.
- 3D optical data storage has also been proposed.

  Light induced magnetization melting in magnetic photoconductors has also been proposed for high-speed low-energy consumption magneto-optical storage.

Computer Software:

- **Types of Software:** System software
  - Application software
  - Utility Software
  - Demoware
  - Shareware
  - Freeware
  - Firmware
  - Free Software.

SYSTEM SOFTWARE: System software refers to the files and programs that make up your computer's operating system. System files include libraries of functions, system services, drivers for printers and other hardware, system preferences, and other configuration files. The programs that are part of the system software include assemblers, compilers, file management tools, system utilites, and debuggers.

The system software is installed on your computer when you install your operating system. You can update the software by running programs such as "Windows Update" for Windows or "Software Update" for Mac OS X. Unlike application programs, however, system software is not meant to be run by the end user. For example, while you might use your Web browser every day, you probably don't have much use for an assembler program (unless, of course, you are a computer programmer).

Since system software runs at the most basic level of your computer, it is called "low-level" software. It generates the user interface and allows the operating system to interact with the hardware. Fortunately, you don't have to worry about what the system software is doing since it just runs in the background. It's nice to think you are working at a "high-level" anyway.

System software includes:

- Operating systems.
- Device drivers.
- Middleware.
- Utility software.
- Shells and windowing systems.

APPLICATION SOFTWARE: An application software (app or application for short) is a computer software designed to perform a group of coordinated functions, tasks, or activities for the benefit of the user. Examples of an application include a word processor, a spreadsheet, an accounting application, a web browser, a media player, an aeronautical flight simulator, a console game or a photo editor. The collective noun application software refers to all applications collectively. This contrasts with system software, which is mainly involved with running the computer.

Applications may be bundled with the computer and its system software or published separately, and may be coded as proprietary, open-source or university projects.[2] Apps built for mobile platforms are called mobile apps.

Application Software: Task Oriented

Application Software Type	**Examples**
Word processing software	MS Word, WordPad and Notepad
Database software	Oracle, MS Access etc
Spreadsheet software	Apple Numbers, Microsoft Excel
Multimedia software	Real Player, Media Player

Utility Software: Utility software is system software designed to help analyze, configure, optimize or maintain a computer. It is used to support the computer infrastructure in contrast to application software, which is aimed at directly performing tasks that benefit ordinary users.

Although a basic set of utility programs is usually distributed with an operating system (OS), utility software is not considered part of the operating system, and users often install replacements or additional utilities. It provides additional facilities to carry out tasks which are beyond the capabilities of the operating system.

Many utilities which might affect the entire computer system require the user to have elevated privileges, while others, which operate only on the user's data, do not.

- **Anti-virus** utilities scan for computer viruses and remove them.
- **Clipboard managers** expand the clipboard functionality of an operating system .

- **Memory testers** check for memory failures.
- **Package managers** are used to configure, install or keep up to date other software on a computer.
- **Registry cleaners** clean and optimize the Windows Registry by removing old registry keys that are no longer in use.
- **Screensavers** were desired to prevent phosphor burn-in on CRT and plasma computer monitors by blanking the screen or filling it with moving images or patterns when the computer is not in use. Contemporary screensavers are used primarily for entertainment or security.
- **Network utilities** analyze the computer's network connectivity, configure network settings, check data transfer or log events.
- **System monitors** monitor resources and performance in a computer system.
- **System profilers** provide detailed information about the software installed and hardware attached to the computer.

Demoware : Demoware is a term of distinction used to differentiate between types of shareware software. Demoware is commercial software released by way of a free download in a version which is limited in one (or more) of the following ways:

- Remains functional for a set number of days from installation.
- May only be launched a set number of times.
- Is limited to a set number of "days of use".
- Program execution terminates after a set time period (typically between 5 and 60 min)
- Limited number of times an important function may be used (eg, CD burn)

Shareware: Shareware is software that is distributed free on a trial basis with the understanding that the user may need or want to pay for it later. Some software developers offer a shareware version of their program with a built-in expiration date (after 30 days, the user can no longer get access to the program).

Shareware is software that you can use on a trial basis before paying for it. Unlike freeware, shareware often has limited functionality or may only be used for a limited time before requiring payment and registration. Once you pay for a shareware program, the program is fully functional and the time limit is removed.

In the 1980s and 1990s, shareware was a popular way for small developers to distribute software. The advent of CDs allowed multiple developers to deliver their software programs as a collection, such as "Top 100 Mac Games." Other shareware collections included utilities, graphics programs, and productivity applications. In many cases, these programs were fully functional and simply requested a donation from users. Programs that incessantly reminded users to register and pay for the software became known as "nagware."

Today, the most common type of shareware programs are trial programs, which are also called "trialware" or "demoware." These programs are provided as demos that you can try for a limited time, such as two weeks or one month. Once the trial period expires, you must pay for the software in order to continue using it. Most shareware demos can be downloaded directly from the software publisher's website.

Freeware : Freeware is software that is available for use at no monetary cost. In other words, while freeware may be used without payment it is most often proprietary software, and usually modification, re-distribution or reverse-engineering without the author's permission is prohibited.Two historic examples of freeware include Skype and Adobe Acrobat Reader. There is no agreed set of rights or a license or an EULA which would define "freeware" unambiguously; every Freeware publisher defines their own rules for their Freeware. For instance, redistribution of Freeware by third-parties is often permitted but there is a significant portion of Freeware which prohibits redistribution.

Freeware, although itself free of charge, may be intended to benefit its producer, e.g. by encouraging sales of a more capable version ("Freemium" or Shareware business model). The source code of freeware is typically not available, unlike free and open source software which are also often distributed free of charge.

Firmware: In electronic systems and computing, firmware is a specific class of computer software that provides the low-level control for the device's specific hardware. Firmware can either provide a standardized operating environment for the device's more complex software (allowing more hardware-independence), or, for less complex devices, act as the device's complete operating system, performing all control, monitoring and data manipulation functions. Typical examples of devices containing firmware are embedded systems, consumer appliances, computers, computer peripherals, and others. Almost all electronic devices beyond the simplest contain some firmware.

Firmware is held in non-volatile memory devices such as ROM, EPROM, or flash memory. Changing the firmware of a device may rarely or never be done during its lifetime; some firmware memory devices are permanently installed and cannot be changed after manufacture. Common reasons for updating firmware include fixing bugs or adding features to the device. This may require ROM integrated circuits to be physically replaced, or flash memory to be reprogrammed through a special procedure.[2] Firmware such as the ROM BIOS of a personal computer may contain only elementary basic functions of a device and may only provide services to higher-level software. Firmware such as the program of an embedded system may be the only program that will run on the system and provide all of its functions.

Before the inclusion of integrated circuits, other firmware devices included a discrete semiconductor diode matrix. The Apollo guidance computer had firmware consisting of a specially manufactured core memory plane, called "core rope memory", where data was stored by physically threading wires through (1) or around (0) the core storing each data bit.

Free software :

Free software or libre software is computer software distributed under terms that allow users to run the software for any purpose as well as to study, change, and distribute it and any adapted versions. Free software is a matter of liberty, not price: users —individually or in cooperation with computer programmers— are free to do what they want with their copies of a free software (including profiting from them) regardless of how much is paid to obtain the program. Computer programs are deemed free insofar as they give users (not just the developer) ultimate control over the first, thereby allowing them to control what their devices are programmed to do.

The right to study and modify a computer program entails that source code —the preferred format for making changes be made available to users of that program. While this is often called 'access to source code' or 'public availability', the Free Software Foundation recommends against thinking in those terms,[10] because it might give the impression that users have an obligation (as opposed to a right) to give non-users a copy of the program.

Although the term free software had been used loosely in the past, Richard Stallman is credited with tying it to the sense under discussion and starting the Free Software movement in 1983, when he launched the GNU Project: a collaborative effort to create a freedom-respecting operating system, and revive the spirit of cooperation once prevalent among hackers during the early days of computing.

Operating System: An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

The dominant desktop operating system is Microsoft Windows with a market share of around 82.74%. macOS by Apple Inc. is in second place (13.23%), and the varieties of Linux are collectively in third place (1.57%). In the mobile (smartphone and tablet combined) sector, use in 2017 is up to 70% of Google's Android[4] and according to third quarter 2016 data, Android on smartphones is dominant with 87.5 percent and a growth rate 10.3 percent per year, followed by Apple's IOS with 12.1 percent and a per year decrease in market share of 5.2 percent, while other operating systems amount to just 0.3 percent. Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems, such as embedded and real-time systems, exist for many applications.

Types of operating systems:

Single- and multi-tasking

A single-tasking system can only run one program at a time, while a multi-tasking operating system allows more than one program to be running in concurrency. This is achieved by time-sharing, where the available processor time is divided between multiple processes. These processes are each interrupted repeatedly in time slices by a task-scheduling subsystem of the operating system. Multi-tasking may be characterized in preemptive and co-operative types. In preemptive multitasking, the operating system slices the CPU time and dedicates a slot to each of the programs. Unix-like operating systems, such as Solaris and Linux—as well as non-Unix-like, such as AmigaOS—support preemptive multitasking. Cooperative multitasking is achieved by relying on each process to provide time to the other processes in a defined manner. 16-bit versions of Microsoft Windows used cooperative multi-tasking. 32-bit versions of both Windows NT and Win9x, used preemptive multi-tasking.

Single- and multi-user

Single-user operating systems have no facilities to distinguish users, but may allow multiple programs to run in tandem.[6] A multi-user operating system extends the basic concept of multi-tasking with facilities that identify processes and resources, such as disk space, belonging to multiple users, and the system permits multiple users to interact with the system at the same time. Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources to multiple users.

Distributed

A distributed operating system manages a group of distinct computers and makes them appear to be a single computer. The development of networked computers that could be linked and communicate with each other gave rise to distributed computing. Distributed computations are carried out on more than one machine. When computers in a group work in cooperation, they form a distributed system.[7]

Templated

In an OS, distributed and cloud computing context, templating refers to creating a single virtual machine image as a guest operating system, then saving it as a tool for multiple running virtual machines. The technique is used both in virtualization and cloud computing management, and is common in large server warehouses.[8]

Embedded

Embedded operating systems are designed to be used in embedded computer systems. They are designed to operate on small machines like PDAs with less autonomy. They are able to operate with a limited number of resources. They are very compact and extremely efficient by design. Windows CE and Minix 3 are some examples of embedded operating systems.

Real-time

A real-time operating system is an operating system that guarantees to process events or data by a specific moment in time. A real-time operating system may be single- or multi-tasking, but when multitasking, it uses specialized scheduling algorithms so that a deterministic nature of behavior is achieved. An event-driven system switches between tasks based on their priorities or external events while time-sharing operating systems switch tasks based on clock interrupts

Library

A library operating system is one in which the services that a typical operating system provides, such as networking, are provided in the form of libraries and composed with the application and configuration code to construct a unikernel: a specialized, single address space, machine image that can be deployed to cloud or embedded environments.

Programming languages:

- Machine
- Assembly
- High Level
- 4 GL.

- **Machine language** is a set of instructions executed directly by a computer's central processing unit(CPU). Each instruction performs a very specific task, such as a load, a jump, or an ALU operation on a unit of data in a CPU register or memory. Every program directly executed by a CPU is made up of a series of such instructions. (The phrase 'directly executed' needs some clarification; machine code is by definition the lowest level of programming detail visible to the programmer, but internally many processors use microcode or optimise and transform machine code instructions into sequences of micro-ops in a sophisticated way.)

- Numerical machine code (i.e., not assembly code) may be regarded as the lowest-level representation of a compiled or assembled computer program or as a primitive and hardware-dependent programming language. While it is possible to write programs directly in numerical machine code, it is tedious and error prone to manage individual bits and calculate numerical addresses and constants manually. For this reason, programs are almost never written directly in machine code in modern contexts. Three exceptions are: for low level debugging, program patching, and assembly language disassembly (which would be called for in the absence of assembly source code, when only the machine language object code is available).

- The overwhelming majority of practical programs today are written in higher-level languages or assembly language. The source code is then translated to executable machine code by utilities such as compilers, assemblers, and linkers, with the important exception of interpreted programs,[1] which are not translated into machine code. However, the *interpreter* itself, which may be seen as an executor or processor, performing the instructions of the source code, typically consists of directly executable machine code (generated from assembly or high-level language source code).

Assembly Language: An **assembly** (or **assembler**) **language**, often abbreviated **asm**, is a low-level programming language, in which there is a very strong (but often not one-to-one) correspondence between the language and the architecture'smachine code instructions. Each assembly language is specific to a particular computer architecture. In contrast, most high-level programming languages are generally portable across multiple architectures but require interpreting or compiling. Assembly language may also be called *symbolic machine code*.

Assembly language is converted into executable machine code by a utility program referred to as an *assembler*. The conversion process is referred to as *assembly*, or *assembling* the source code. *Assembly time* is the computational step where an assembler is run.

Assembly language uses a mnemonic to represent each low-level machine instruction or opcode, typically also each architectural register, flag, etc. Many operations require one or more operands in order to form a complete instruction. Most assemblers can take expressions of numbers, named constants, registers, and labels as operands. Thus, the programmers are freed from tedious repetitive calculations. Depending on the architecture, these elements may also be combined for specific instructions or addressing modes using offsets or other data as well as fixed addresses. Many assemblers offer additional mechanisms to facilitate program development, to control the assembly process, and to aid debugging.

High-Level Programming Language

In computer science, a **high-level programming language** is a programming language with strong abstraction from the details of the computer. In comparison to low-level programming languages, it may use natural language *elements*, be easier to use, or may automate (or even hide entirely) significant areas of computing systems (e.g. memory management), making the process of developing a program simpler and more understandable relative to a lower-level language. The amount of abstraction provided defines how "high-level" a programming language is.

In the 1960s, low-level programming languages using a compiler were commonly called **autocodes**. Examples of autocodes are COBOL and Fortran.

The first high-level programming language designed for computers was Plankalkül, created by Konrad Zuse. However, it was not implemented in his time, and his original contributions were (due to World War II) largely isolated from other developments, although it influenced Heinz Rutishauser's language "Superplan" (and to some degree also Algol). The first really widespread high-level language was Fortran, a machine independent development of IBM's earlier Autocode systems. Algol, defined in 1958 and 1960, by committees of European and American computer scientists, introduced recursion as well as nested functions under lexical scope. It was also the first language with a clear distinction between value and name-parameters and their corresponding semantics.Algol also introduced several structured programming concepts, such as the **while-do** and **if-then-else** constructs and its syntax was the first to be described by a formal method, Backus–Naur form (BNF). During roughly the same period Cobol introduced records (also called structs) and Lisp introduced a fully general lambda abstraction in a programming language for the first time.

4 GL: A 4th-generation programming language (4GL) or (procedural language) is any computer programming language that belongs to a class of languages envisioned as an advancement upon third-generation programming languages (3GL). Each of the programming language generations aims to provide a higher level of abstraction of the internal computer hardware details, making the language more programmer-friendly, powerful and versatile. While the definition of 4GL has changed over time, it can be typified by operating more with large collections of information at once rather than focusing on just bits and bytes. Languages claimed to be 4GL may include support for database management, report generation, mathematical optimization, GUI development, or web development. Some researchers state that 4GLs are a subset of domain-specific languages.

The concept of 4GL was developed from the 1970s through the 1990s, overlapping most of the development of 3GL. While 3GLs like C, C++, C#, Java, and JavaScript remain popular for a wide variety of uses, 4GLs as originally defined found narrower uses.[citation needed] Some advanced 3GLs like Python, Ruby, and Perl combine some 4GL abilities within a general-purpose 3GL environment. Also, libraries with 4GL-like features have been developed as add-ons for most popular 3GLs. This has blurred the distinction of 4GL and 3GL.

Data Representation In Computers:

Are Computers mystery machines? We have all seen computers do seemingly miraculous things with all kinds of sounds, pictures, graphics, numbers, and text. It seems we can build a replica of parts of our world inside the computer. You might think that this amazing machine is also amazingly complicated - it really is not.

In fact, all of the wonderful multi-media that we see on modern computers is all constructed from simple ON/OFF switches - millions of them - but really nothing much more complicated than a switch. The trick is to take all of the real-world sound, picture, number etc. data that we want in the computer and convert it into the kind of data that can be represented in switches.

Like with the artist’s abstract composition, the trick is to take all of the real-world sound, picture, number, etc. data that we want in the computer and convert it into the kind of data that can be represented in switches.

Computers Are Electronic Machines. The computer uses electricity, not mechanical parts, for its data processing and storage. Electricity is plentiful, moves very fast through wires, and electrical parts fail much less frequently than mechanical parts. The computer does have some mechanical parts, like its disk drive, (which are often the sources for computer failures), but the internal data processing and storage is electronic, which is fast and reliable (as long as the computer is plugged in).

Electricity can flow through switches: if the switch is closed, the electricity flows; if the switch is open, the electricity does not flow. To process real-world data in the computer, we need a way to represent the data in switches. Computers do this representation using a binary coding system.

Number System of computers:

- Binary
- Octal
- Hexa Decimal

Binary:

In mathematics and digital electronics, a binary number is a number expressed in the base-2 numeral system or binary numeral system, which uses only two symbols: typically 0 (zero) and 1 (one).

The base-2 numeral system is a positional notation with a radix of 2. Each digit is referred to as a bit. Because of its straightforward implementation in digital electronic circuitry using logic gates, the binary system is used by almost all modern computers and computer-based devices.

Octal:

The **octal** numeral system, or **oct** for short, is the base-8 number system, and uses the digits 0 to 7. Octal numerals can be made from binary numerals by grouping consecutive binary digits into groups of three (starting from the right). For example, the binary representation for decimal 74 is 1001010. Two zeroes can be added at the left: (00)1 001 010, corresponding the octal digits 1 1 2, yielding the octal representation 112.

In the decimal system each decimal place is a power of ten.

Hexa Decimal:

In mathematics and computing, **hexadecimal** (also **base 16**, or **hex**) is a positional numeral system with a radix, or base, of 16. It uses sixteen distinct symbols, most often the symbols **0**–**9** to represent values zero to nine, and **A**–**F** (or alternatively **a**–**f**) to represent values ten to fifteen.

Hexadecimal numerals are widely used by computer system designers and programmers, as they provide a more human-friendly representation of binary-coded values. Each hexadecimal digit represents four binary digits, also known as a nibble, which is half a byte. For example, a single byte can have values ranging from 0000 0000 to 1111 1111 in binary form, which can be more conveniently represented as 00 to FF in hexadecimal.

In mathematics, a subscript is typically used to specify the radix. For example the decimal value 10,995 would be expressed in hexadecimal as $2AF3_{16}$. In programming, a number of notations are used to support hexadecimal representation, usually involving a prefix or suffix. The prefix `0x` is used in C and related languages, which would denote this value by 0x2AF3.

Coding System:

- **ASCII**
- **BCD**
- **EBCDIC**

ASCII :

American Standard Code for Information Interchange, is a character encoding standard for electronic communication. ASCII codes represent text in computers, telecommunications equipment, and other devices. Most modern character-encoding schemes are based on ASCII, although they support many additional characters.

ASCII is the traditional name for the encoding system; the Internet Assigned Numbers Authority (IANA) prefers the updated name **US-ASCII**, which clarifies that this system was developed in the US and based on the typographical symbols predominantly in use there.[2]

ASCII is one of a 1963 List of IEEE milestones.

BCD :

In the coding, when numbers, letters or words are represented by a specific group of symbols, it is said that the number, letter or word is being encoded. The group of symbols is called as a code. The digital data is represented, stored and transmitted as group of binary bits. This group is also called as **binary code**. The binary code is represented by the number as well as alphanumeric letter.

Advantages of Binary Code

Following is the list of advantages that binary code offers.

- Binary codes are suitable for the computer applications.
- Binary codes are suitable for the digital communications.
- Binary codes make the analysis and designing of digital circuits if we use the binary codes.
- Since only 0 & 1 are being used, implementation becomes easy.

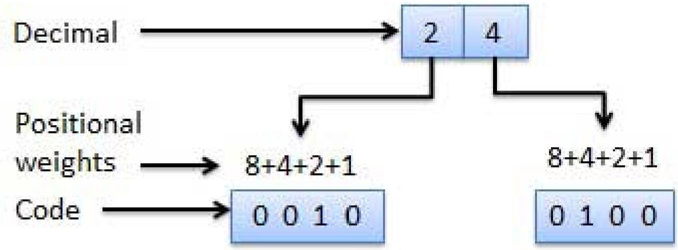
Classification of binary codes

The codes are broadly categorized into following four categories.

- Weighted Codes
- Non-Weighted Codes
- Binary Coded Decimal Code
- Alphanumeric Codes
- Error Detecting Codes
- Error Correcting Codes

Weighted Codes:

Weighted binary codes are those binary codes which obey the positional weight principle. Each position of the number represents a specific weight. Several systems of the codes are used to express the decimal digits 0 through 9. In these codes each decimal digit is represented by a group of four bits.

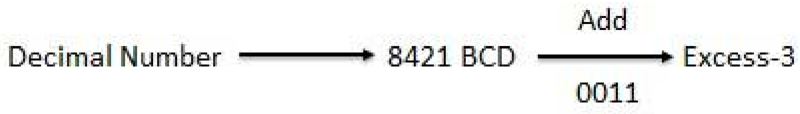


Non-Weighted Codes:

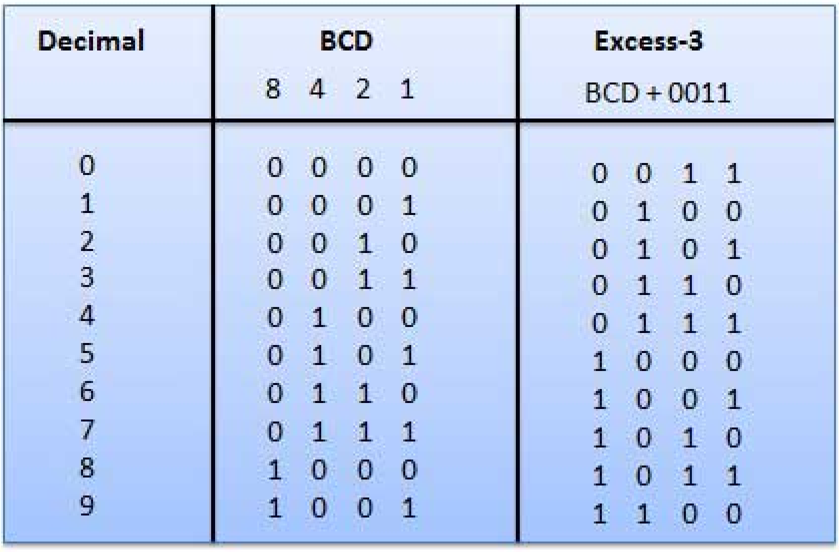
In this type of binary codes, the positional weights are not assigned. The examples of non-weighted codes are Excess-3 code and Gray code.

Excess-3 code:

The Excess-3 code is also called as XS-3 code. It is non-weighted code used to express decimal numbers. The Excess-3 code words are derived from the 8421 BCD code words adding (0011)2 or (3)10 to each code word in 8421. The excess-3 codes are obtained as follows −



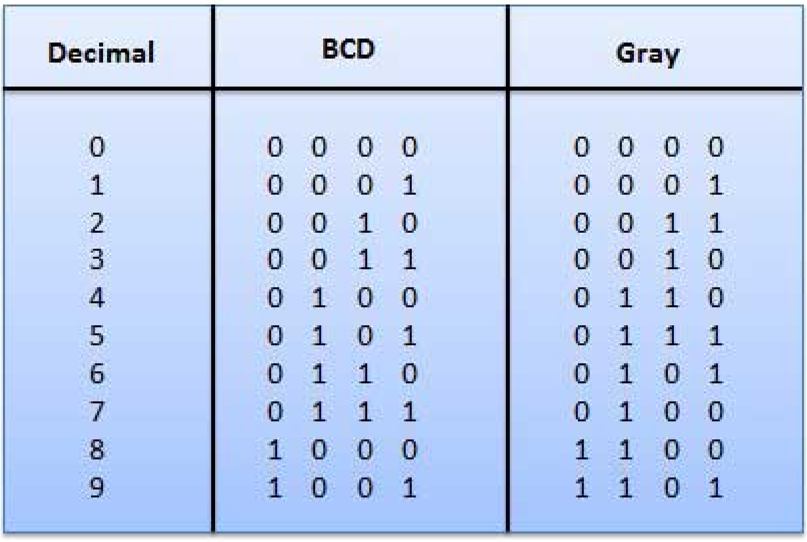
Example



Decimal	BCD	Excess-3
	8 4 2 1	BCD + 0011
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0

Gray Code:

It is the non-weighted code and it is not arithmetic codes. That means there are no specific weights assigned to the bit position. It has a very special feature that, only one bit will change each time the decimal number is incremented as shown in fig. As only one bit changes at a time, the gray code is called as a unit distance code. The gray code is a cyclic code. Gray code cannot be used for arithmetic operation.



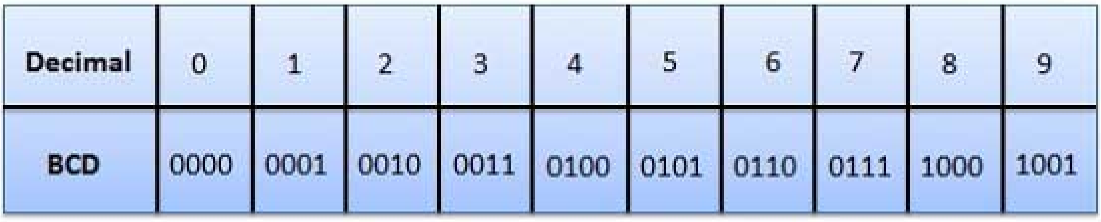
Decimal	BCD	Gray
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1

Application of Gray code:

- Gray code is popularly used in the shaft position encoders.
- A shaft position encoder produces a code word which represents the angular position of the shaft.

Binary Coded Decimal (BCD) code

In this code each decimal digit is represented by a 4-bit binary number. BCD is a way to express each of the decimal digits with a binary code. In the BCD, with four bits we can represent sixteen numbers (0000 to 1111). But in BCD code only first ten of these are used (0000 to 1001). The remaining six code combinations i.e. 1010 to 1111 are invalid in BCD.



Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Advantages of BCD Codes

- It is very similar to decimal system.
- We need to remember binary equivalent of decimal numbers 0 to 9 only.

Disadvantages of BCD Codes

- The addition and subtraction of BCD have different rules.
- The BCD arithmetic is little more complicated.
- BCD needs more number of bits than binary to represent the decimal number. So BCD is less efficient than binary.

Alphanumeric codes;

A binary digit or bit can represent only two symbols as it has only two states '0' or '1'. But this is not enough for communication between two computers because there we need many more symbols for communication. These symbols are required to represent 26 alphabets with capital and small letters, numbers from 0 to 9, punctuation marks and other symbols.

The alphanumeric codes are the codes that represent numbers and alphabetic characters. Mostly such codes also represent other characters such as symbol and various instructions necessary for conveying information. An alphanumeric code should at least represent 10 digits and 26 letters of alphabet i.e. total 36 items. The following three alphanumeric codes are very commonly used for the data representation.

- American Standard Code for Information Interchange (ASCII).
- Extended Binary Coded Decimal Interchange Code (EBCDIC).
- Five bit Baudot Code.

ASCII code is a 7-bit code whereas EBCDIC is an 8-bit code. ASCII code is more commonly used worldwide while EBCDIC is used primarily in large IBM computers.

Error Codes

There are binary code techniques available to detect and correct data during data transmission.

Error Code	Description
Error Detection and Correction	Error detection and correction code techniques

EBCDIC:

EBCDIC was devised in 1963 and 1964 by IBM and was announced with the release of the IBM System/360 line of mainframe computers. It is an eight-bit character encoding, developed separately from the seven-bit ASCII encoding scheme. It was created to extend the existing Binary-Coded Decimal (BCD) Interchange Code, or BCDIC, which itself was devised as an efficient means of encoding the two *zone* and *number* punches on punched cards into six bits. The distinct encoding of 's' and 'S' (using position 2 instead of 1) was maintained from punched cards where it was desirable not to have hole punches too close to each other to ensure the integrity of the physical card.

While IBM was a chief proponent of the ASCII standardization committee,[3] the company did not have time to prepare ASCII peripherals (such as card punch machines) to ship with its System/360 computers, so the company settled on EBCDIC.[2] The System/360 became wildly successful, together with clones such as RCA Spectra 70, ICL System 4, and Fujitsu FACOM, thus so did EBCDIC.

All IBM mainframe and midrange peripherals and operating systems use EBCDIC as their inherent encoding[4] (with toleration for ASCII, for example, ISPF in z/OS can browse and edit both EBCDIC and ASCII encoded files). Software and many hardware peripherals can translate to and from encodings, and modern mainframes (such as IBM zSeries) include processor instructions, at the hardware level, to accelerate translation between character sets.

There is an EBCDIC-oriented Unicode Transformation Format called UTF-EBCDIC proposed by the Unicode consortium, designed to allow easy updating of EBCDIC software to handle Unicode, but not intended to be used in open interchange environments. Even on systems with extensive EBCDIC support, it has not been popular. For example, z/OS supports Unicode (preferring UTF-16 specifically), but z/OS only has limited support for UTF-EBCDIC.

IBM AIX running on the RS/6000 and its descendants including the IBM Power Systems, Linux running on z Systems, and operating systems running on the IBM PCand its descendants use ASCII, as did AIX/370 and AIX/390 running on System/370 and System/390 mainframe.

Compatibility with ASCII

The fact that all the code points were different was less of a problem for inter-operating with ASCII than the fact that sorting EBCDIC put lowercase letters before uppercase letters and letters before numbers, exactly the opposite of ASCII.

Software portability and data exchange are hindered by EBCDIC's lack of codes for several symbols (such as the brace characters) commonly used in programming and in network communications.

The gaps between letters made simple code that worked in ASCII fail on EBCDIC. For example, "for (c='A';c<='Z';++c)" would set c to the 26 letters in the ASCII alphabet, but 41 characters including a number of unassigned ones in EBCDIC. Fixing this required complicating the code with function calls which was greatly resisted by programmers.

All ASCII codes stored within an eight-bit byte had nonnegative values on systems such as the PDP-11 that treated bytes as signed quantities. Software on those platforms often took advantage of that property, causing problems when it was ported to EBCDIC-based environments where many character codes had a 1 as the "sign" bit.

By using all eight bits EBCDIC may have encouraged the use of the eight-bit byte by IBM, while ASCII was more likely to be adopted by systems with 36 bits (as five seven-bit ASCII characters fit into one word). As eight-bit bytes became widespread, ASCII systems sometimes used the "unused" bit for other purposes such as parity, thus making it more difficult to transition to larger character sets.

Computer Viruses:

A computer virus is a type of malicious software program ("malware") that, when executed, replicates itself by modifying other computer programs and inserting its own code.[1] When this replication succeeds, the affected areas are then said to be "infected" with a computer virus.

Virus writers use social engineering deceptions and exploit detailed knowledge of security vulnerabilities to initially infect systems and to spread the virus. The vast majority of viruses target systems running Microsoft Windows,[employing a variety of mechanisms to infect new hosts ,and often using complex anti-detection/stealth strategies to evade antivirus software. Motives for creating viruses can include seeking profit (e.g., with ransomware), desire to send a political message, personal amusement, to demonstrate that a vulnerability exists in software, for sabotage and denial of service, or simply because they wish to explore cyber security issues, artificial life and evolutionary algorithms.

Computer viruses currently cause billions of dollars' worth of economic damage each year ,due to causing system failure, wasting computer resources, corrupting data, increasing maintenance costs, etc. In response, free, open-source antivirus tools have been developed, and an industry of antivirus software has cropped up, selling or freely distributing virus protection to users of various operating systems. As of 2005, even though no currently existing antivirus software was able to uncover all computer viruses (especially new ones), computer security researchers are actively searching for new ways to enable antivirus solutions to more effectively detect emerging viruses, before they have already become widely distributed.

The term "virus" is also commonly, but erroneously, used to refer to other types of malware. "Malware" encompasses computer viruses along with many other forms of malicious software, such as computer "worms", ransomware, spyware, adware, trojan horses, keyloggers, rootkits, bootkits, malicious Browser Helper Object (BHOs) and other malicious software. The majority of active malware threats are actually trojan horse programs or computer worms rather than computer viruses. The term computer virus, coined by Fred Cohen in 1985, is a misnomer.Viruses often perform some type of harmful activity on infected host computers, such as acquisition of hard disk space or central processing unit (CPU) time, accessing private information (e.g., credit card numbers), corrupting data, displaying political or humorous messages on the user's screen, spamming their e-mail contacts, logging their keystrokes, or even rendering the computer useless. However, not all viruses carry a destructive "payload" and attempt to hide themselves—the defining characteristic of viruses is that they are self-replicating computer programs which modify other software without user consent.

Graphic Designing:

- **Corel Draw**
- **Photoshop**
- **Pagemaker**

Corel Draw : CorelDraw (styled Corel DRAW) is a vector graphics editor developed and marketed by Corel Corporation. It is also the name of Corel's Graphics Suite, which bundles CorelDraw with bitmap-image editor Corel Photo-Paint as well as other graphics-related programs (see below). The latest version is marketed as CorelDraw Graphics Suite 2018 (equivalent to version 20), and was released in April 10, 2018[1]. CorelDraw is designed to edit two-dimensional images such as logos and posters.

In 1987, Corel engineers Michel Bouillon and Pat Beirne undertook to develop a vector-based illustration program to bundle with their desktop publishing systems. That program, CorelDraw, was initially released in 1989. CorelDraw 1.x and 2.x ran under Windows 2.x and 3.0. CorelDraw 3.0 came into its own with Microsoft's release of Windows 3.1. The inclusion of TrueType in Windows 3.1 transformed CorelDraw into a serious illustration program capable of using system-installed outline fonts without requiring third-party software such as Adobe Type Manager; paired with a photo-editing program (Corel Photo-Paint), a font manager and several other pieces of software, it was also part of the first all-in-one graphics suite.

Supported platforms:

CorelDraw was originally developed for Microsoft Windows 3 and currently runs on Windows XP, Windows Vista, Windows 7, Windows 8 and Windows 10 .[31] The latest version, 2017, was released on 11 April 2017.

Versions for Mac OS and Mac OS X were at one time available, but due to poor sales these were discontinued. The last port for Linux was version 9 (released in 2000, it did not run natively; instead, it used a modified version of Wine to run) and the last version for OS X was version 11 (released in 2001). Also, up until version 5, CorelDraw was developed for Windows 3.1x, CTOS and OS/2.

With version 6, CorelDraw introduced the automation of tasks using a Corel proprietary scripting language, COREL Script. With version 10, support for VBA (Visual Basic for Applications) was introduced for scripting by what Corel calls now macros. Corel recommends to no longer use the COREL Script language but only VBA.

Structure

In its first versions, the CDR file format was a completely proprietary file format primarily used for vector graphic drawings, recognizable by the first two bytes of the file being "WL". Starting with CorelDraw 3, the file format changed to a Resource Interchange File Format

(RIFF) envelope, recognizable by the first four bytes of the file being "RIFF", and a "CDR*vrsn" in bytes 9 to 15, with the asterisk "*" being in early versions just a blank. Beginning with CorelDraw 4 it included the version number of the writing program in hexadecimal ("4" meaning version 4, "D" meaning version 14). The actual data chunk of the RIFF remains a Corel proprietary format.

From version X4 (14) on, the CDR file is a ZIP-compressed directory of several files, among them XML-files and the RIFF-structured riffdata.cdr with the familiar version signature in versions X4 (CDREvrsn) and X5 (CDRFvrsn), and a root.dat with CorelDraw X6, where the bytes 9 to 15 look slightly different -- "CDRGfver" in a file created with X6. "F" was the last valid hex digit, and the "fver" now indicates that the letter before does no longer stand for a hex digit.

There is no publicly available CDR file format specification.

Other CorelDraw file formats include CorelDraw Compressed (CDX), CorelDraw Template (CDT)[36] and Corel Presentation Exchange (CMX).

Use of CDR-files in other programs:

In December 2006 the sK1 open source project team started to reverse-engineer the CDR format.[38] The results and the first working snapshot of the CDR importer were presented at the Libre Graphics Meeting 2007 conference taking place in May 2007 in Montreal (Canada).[39] Later on the team parsed the structure of other Corel formats with the help of the open source CDR Explorer.[40] As of 2008, the sK1 project claims to have the best import support for CorelDraw file formats among open source software programs. The sK1 project developed also the Uni Convertor, a command line open source tool which supports conversion from CorelDraw ver.7-X4 formats (CDR/CDT/CCX/CDRX/CMX) to other formats. UniConvertor is also used in Inkscape and Scribus open source projects as an external tool for CorelDraw files importing.

In 2007, Microsoft blocked CDR file format in Microsoft Office 2003 with the release of Service Pack 3 for Office 2003.[44][45] Microsoft later apologized for inaccurately blaming the CDR file format and other formats for security problems in Microsoft Office and released some tools for solving this problem.

In 2012 the joint Libre Office/re-lab team implemented libcdr, a library for reading CDR files from v1 to the currently latest X7 version and CMX files.The library has extensive support for shapes and their properties, including support for color management and spot colors, and has a basic support for text.[48] The library provides a built-in converter to SVG, and a converter to Open Document is provided by writer perfect package. The libcdr library is used in Libre Office starting from version 3.6,[49] and thanks to public API it can be freely used by other applications.

Other applications supporting CDR files:

Main article: Comparison of vector graphics editors

CDR file format import is partially or fully supported in following applications:

- Adobe Illustrator - CorelDraw 5, 6, 7, 8, 9, 10
- Corel Paint Shop Photo Pro
- Corel WordPerfect Office
- Inkscape with Uni Convertor installed; partial support
- Libre Office with lib cdr installed - CorelDraw 1 to X7
- Adobe FreeHand - CorelDraw 7, 8
- Microsoft Visio 2002 - CorelDraw! drawing file versions 3.0, 4.0, 5.0, 6.0 and 7.0 (.cdr), Corel Clipart (.cmx) sK1 - partial support
- Xara Designer Pro and Xara Photo & Graphic Designer - early versions of CorelDraw CDR and CMX.

Photoshop :

Photoshop is a raster graphics editor developed and published by Adobe Systems for macOS and Windows .Photoshop was created in 1988 by Thomas and John Knoll. Since then, it has become the de facto industry standard in raster graphics editing, such that the word "photoshop" has become a verb as in "to Photoshop an image," "photoshopping" and "photoshop contest", though Adobe discourages such use. It can edit and compose raster images in multiple layers and supports masks, alpha compositing and several color models including RGB, CMYK, CIELAB, spot color and duotone. Photoshop has vast support for graphic file formats but also uses its own PSD and PSB file formats which support all the aforementioned features. In addition to raster graphics, it has limited abilities to edit or render text, vector graphics (especially through clipping path), 3D graphics and video. Photoshop's feature set can be expanded by Photoshop plug-ins, programs developed and distributed independently of Photoshop that can run inside it and offer new or enhanced features.

Photoshop's naming scheme was initially based on version numbers. However, in October 2002, following the introduction of Creative Suite branding, each new version of Photoshop was designated with "CS" plus a number; e.g., the eighth major version of Photoshop was Photoshop CS and the ninth major version was Photoshop CS2. Photoshop CS3 through CS6 were also distributed in two different editions: Standard and Extended. In June 2013, with the introduction of Creative Cloud branding, Photoshop's licensing scheme was changed to that of software as a service rental model and the "CS" suffixes were replaced with "CC". Historically, Photoshop was bundled with additional software such as Adobe ImageReady, Adobe Fireworks, Adobe Bridge, Adobe Device Central and Adobe Camera RAW.

Alongside Photoshop, Adobe also develops and publishes Photoshop Elements, Photoshop Lightroom, Photoshop Express and Photoshop Touch. Collectively, they are branded as "The Adobe Photoshop Family". It is currently a licensed software.

PAGEMAKER:

PageMaker was one of the first desktop publishing programs, introduced in 1985 by Aldus on the Apple Macintosh. The combination of PageMaker using the Mac's graphical user interface for document creation and the Apple LaserWriter for output represented the starting point of what became the desktop publishing revolution in the late 1980s. Ported to PCs running Windows 1.0 in 1987,PageMaker helped to popularize the Macintosh platform and the Windows environment.

A key aspect of PageMaker's success was its native support for Adobe Systems' PostScript page description language. Adobe purchased Aldus, and PageMaker, in 1994. The program remained a major force in the high-end DTP market through the early 1990s, but new features were slow in coming. By the mid-1990s, it faced increasing competition from QuarkXPress on the Mac, and to a lesser degree, Ventura on the PC, and by the end of the decade it was no longer a major force. Quark proposed buying the product and cancelling it, but instead, in 1999 Adobe released their "Quark Killer", Adobe In Design. The last major release of PageMaker was in 2001, and customers were offered In Design licenses at a lower cost.

Operating System and Application Program:

An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs.Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,[1][2] although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

The dominant desktop operating system is Microsoft Windows with a market share of around 82.74%. mac OS by Apple Inc. is in second place (13.23%), and the varieties of Linux are collectively in third place (1.57%).In the mobile (smartphone and tablet combined) sector, use in 2017 is up to 70% of Google's Android[4] and according to third quarter 2016 data, Android on smartphones is dominant with 87.5 percent and a growth rate 10.3 percent per year, followed by Apple's IOS with 12.1 percent and a per year decrease in market share of 5.2 percent, while other operating systems amount to just 0.3 percent. Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems, such as embedded and real-time systems, exist for many applications.

Disk Operating System (DOS)

- **DOS basics**
- **Basic DOS Commands**

DOS basics: DOS commands are the commands available in MS-DOS that are used to interact with the operating system and other command line based software. Unlike in Windows, DOS commands are the primary way in which you use the operating system. Windows and other modern OSs use a graphics-based system designed for touch or a mouse.

DOS Commands in Windows: If you use Windows (like Windows 10, 8, 7, etc.) then you have no need for DOS commands because you don't have MS-DOS. The commands in Windows are available from the Command Prompt and are called Command Prompt commands or CMD commands, but they are not DOS commands. Instead, check out my List of Windows CMD Commands for all of the command-line options you have available to you in Windows. I also created a command comparison table to show which commands are available in different Microsoft operating systems. Below is a complete list of the nearly 100 MS-DOS commands, commonly referred to as just DOS commands, available as of MS-DOS 6.22.

Windows

Windows

- ✓ Windows concepts
- ✓ Features
- ✓ windows structure
- ✓ desktop, taskbar
- ✓ start menu
- ✓ my computer
- ✓ Recycle Bin.

Window:

A **window** is an opening in a wall, door, roof or vehicle that allows the passage of light, sound, and air. Modern windows are usually glazed or covered in some other transparent or translucent material, a sash set in a frame[1] in the opening; the sash and frame are also referred to as a window.[2] Many glazed windows may be opened, to allow ventilation, or closed, to exclude inclement weather. Windows often have a latch or similar mechanism to lock the window shut or to hold it open by various amounts.

Types include the eyebrow window, fixed windows, single-hung and double-hung sash windows, horizontal sliding sash windows, casement windows, awning windows, hopper windows, tilt and slide windows (often door-sized), tilt and turn windows, transom windows, sidelight windows, jalousie or louvered windows, clerestory windows, skylights, roof windows, roof lanterns, bay windows, oriel windows, thermal, or Diocletian, windows, picture windows, emergency exit windows, stained glasswindows, French windows, panel windows, and double - and triple paned windows.

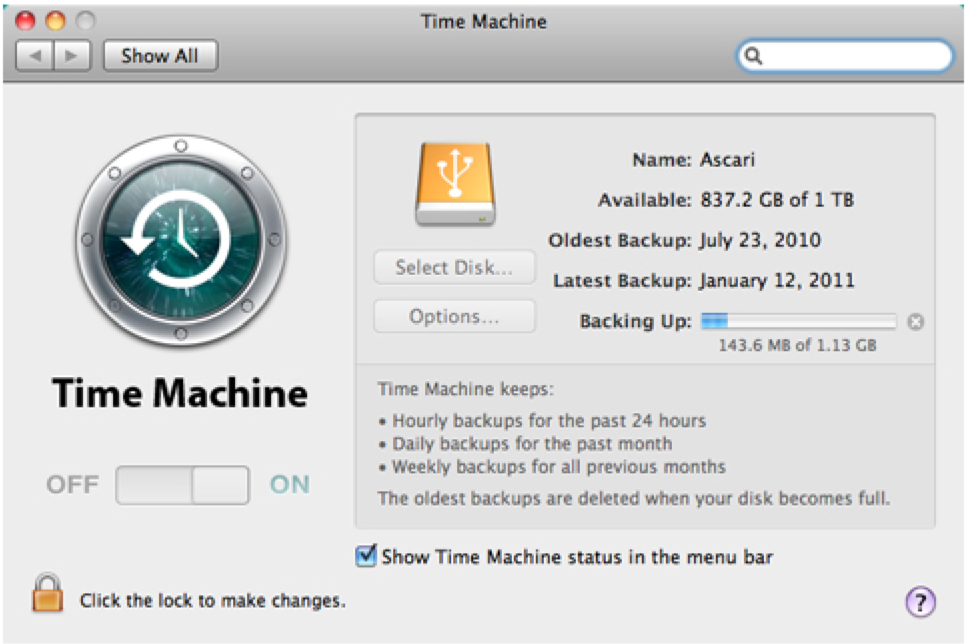
The Romans were the first known to use glass for windows, a technology likely first produced in Roman Egypt, in Alexandriaca. 100 AD. Paper windows were economical and widely used in ancient China, Korea and Japan. In England, glass became common in the windows of ordinary homes only in the early 17th century whereas windows made up of panes of flattened animal horn were used as early as the 14th century. In the 19th century American west, greased paper windows came to be used by itinerant groups.

Window feature:

An overlapped window is a top-level window that has a title bar, border, and client area; it is meant to serve as an application's main window. It can also have a window menu, minimize and maximize buttons, and scroll bars. An overlapped window used as a main window typically includes all of these components.

Each OS has some nugget that we can enjoy, learn from and build on. So here, in no particular order, are 10 different features I love in 10 different OSes.

Mac OS X, Time Machine:



Name: Ascari

Available: 837.2 GB of 1 TB

Oldest Backup: July 23, 2010

Latest Backup: January 12, 2011

Backing Up: 143.6 MB of 1.13 GB

Time Machine keeps:

- Hourly backups for the past 24 hours
- Daily backups for the past month
- Weekly backups for all previous months

The oldest backups are deleted when your disk becomes full.

☑ Show Time Machine status in the menu bar

Click the lock to make changes.

Configuring backups has, traditionally, been one of the least fun things about computing. It's perhaps only slightly less frustrating than trying to recover your system from said backup. If you don't have too many files to back up, services like Dropbox, Sugarsync, and Windows Live Mesh work quite well. In fact, for several years I used Live Sync (formerly Foldershare, now called Live Mesh) to create real-time offsite backups of my most important files. But you can't back up and recover an entire system that way.

Apple introduced Time Machine backup software with Mac OS X 10.5 in 2007, and I have to say it's one of the more brilliant tools I've used. Time Machine is easy to configure and pretty much operates as a set-and-forget service. You can back up to a local drive connected via USB or Firewire or even to network storage via Ethernet or WiFi. As long as your backup volume is available, Time Machine creates hourly, daily and weekly incremental backups of your system.

When trouble strikes, you can go into the Time Machine and recover previous versions of individual files or even the entire system. It's not perfect, but so far I've been successful in all of my attempts at recovery with Time Machine. I've even used Time Machine backups to restore all of a user's files from an older, failing machine to a new one.

Unix, The Shell Terminal

The terminal was my first experience of computing...and by terminal I mean a teleprinter terminal: typewriter keys and a continuous roll of paper scrolling up line by line. The shell was also, for a long time, my portal to the Internet. Then there was MS-DOS...

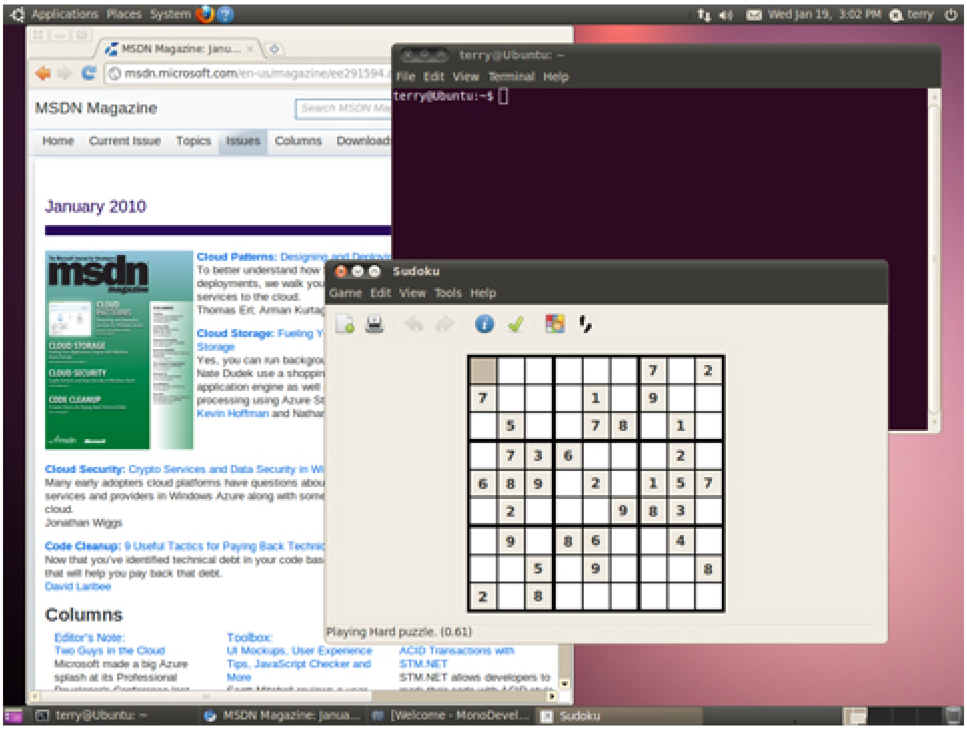
Command line computing lives on, and is even making something of a comeback among users of graphical UI operating systems. Linux and Mac OS X still have their terminal fanboys. And of course you've got a selection of shells, from the original Bourne shell to bash, C, dash, Korn and Z, not to mention fish, psh, rc, scsh, wish and zoidberg.

Windows had its roots in the MS-DOS command line, and continues to this day with the "DOS box" command prompt, cmd.exe. There's Windows PowerShell if you want a more robust scripting environment in Windows, and Cygwin if you prefer something more in line with the traditional Unix terminal.

There's always tension between command-line and graphical interfaces, and for the last decade or more, GUIs have been the dominant face of most OSes. But as Max Steenbergen writes in his article "Commands Lines: Alive & Kicking" for UX Magazine, the command line is making a comeback via app launchers like Alfred, Launchy and GNOME Do. Even applications like Google Chrome and Wolfram|Alpha are blurring the line between textual search and command-line scripting.

Bringing the command line full-circle, a clever coder even built a personal Web site that hosts a command line in the browser window. Retro, or a step into the future?

Ubuntu, Simplified Linux Setup


Applications Places System
Wed Jan 19, 3:02 PM terry
MSDN Magazine: Janu...
msdn.microsoft.com/en-us/magazine/ee291594
MSDN Magazine
Search MSDN Ma
Home Current Issue Topics Issues Columns Download
terry@Ubuntu: ~
File Edit View Terminal Help
Sudoku
Game Edit View Tools Help
terry@Ubuntu: ~
MSDN Magazine: Janua...
[Welcome - MonoDevel...
Sudoku

```
terry@Ubuntu:~$
```

January 2010

Cloud Patterns: Designing and Deployin
To better understand how
deployments, we walk you
services to the cloud.
Thomas Erl; Arman Kurtag

Cloud Storage: Fueling Y
Storage
Yes, you can run backgrou
Nate Dudek use a shoppin
application engine as well
processing using Azure St
Kevin Hoffman and Nathan

Cloud Security: Crypto Services and Data Security in Wi
Many early adopters cloud platforms have questions abou
services and providers in Windows Azure along with some
cloud.
Jonathan Wiggs

Code Cleanup: 9 Useful Tactics for Paying Back Technic
Now that you've identified technical debt in your code bas
that will help you pay back that debt.
David Laribee

Columns

Editor's Note:
Two Guys in the Cloud
Microsoft made a big Azure
splash at its Professional

Toolbox:
UI Mockups, User Experience
Tips, JavaScript Checker and
More

ACID Transactions with
STM.NET
STM.NET allows developers to

						7		2
7				1		9		
	5			7	8		1	
	7	3	6				2	
6	8	9		2		1	5	7
	2				9	8	3	
	9		8	6			4	
		5		9				8
2		8						

Playing Hard puzzle. (0.61)

Much of the Linux revolution has been powered by hackers of the first order. While getting a Linux system up and running isn't rocket science, it does take quite a bit of planning (is my hardware compatible?), knowledge (sudo what?) and time (I've got work to do...). Of course, once you figure those details out, you end up with a powerful, highly customizable and secure system that runs well even on modest hardware.

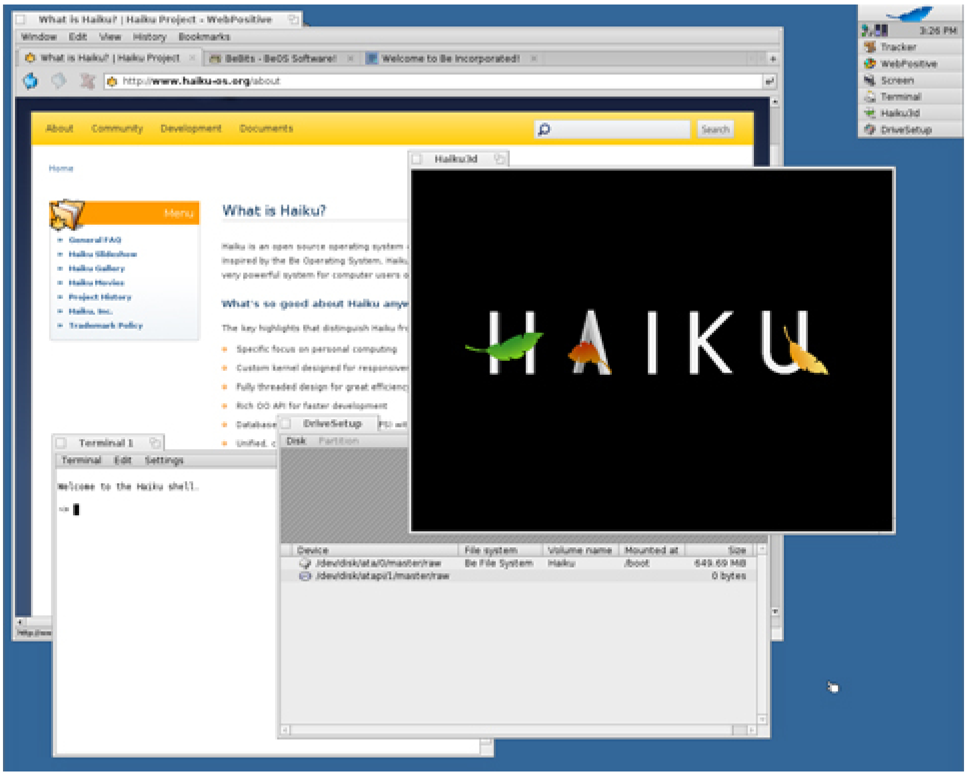
Over the years, increasingly easy access to configuration and installation information via the internet has helped Linux reach a broader audience . A large and growing list of high-quality, free, open-source software for Linux also contributes to its appeal. Still, the learning curve has been steep and the availability of over two hundred different Linux distros makes the choice of where to start difficult.

That is until the release of the Debian-derived Ubuntu in 2004. Ubuntu aims for easy installation and configuration, and that's been my experience so far. You can download a live CD ISO or a Windows installer to get going. It doesn't require much of a commitment if you just want to give Ubuntu a try. Burn the ISO to CD and boot from that, or install it in a virtual machine using VirtualBox, Virtual PC or VMWare Player.

The Ubuntu installation includes a lot of software, so you can start playing or working with right away: e-mail and chat apps, Firefox and Chrome browsers, media apps and OpenOffice, among many others. And of course, Linux offers a cornucopia of tools for the developer. The

Ubuntu Software Center gives you one-click access to a huge library of apps, and updating your software is simple and automatic (and much less intrusive than Windows Update).

BeOS, 64-Bit Journaling File System:



What is Haiku?

Haiku is an open source operating system
inspired by the Be Operating System. Haiku
very powerful system for computer users

What's so good about Haiku anyw

The key highlights that distinguish Haiku fr

- Specific focus on personal computing
- Custom kernel designed for responsive
- Fully threaded design for great efficienc
- Rich OO API for faster development
- Database
- Unified, c

```
Welcome to the Haiku shell.

~>
```

Device	File system	Volume name	Mounted at	Size
/dev/disk/ata/0/master/raw	Be File System	Haiku	/boot	649.69 MB
/dev/disk/atapi/1/master/raw				0 bytes

When Jean Louis Gasse left Apple, he founded a new team that created the charming and forward-looking BeOS in 1991. At the time, BeOS featured some pretty radical technology. Designed from the ground up as an efficient, lightweight multithreaded system with preemptive multitasking, it was very fast on modest hardware and scaled up to take advantage of any processors on the system (in those days, rarely more than two, but still...).

The file system included with BeOS, however, is one of its truly cool features. Called BFS (BeOS File System), it was a 64-bit journaling file system using file attributes, or metadata. The ability to query and sort against file metadata gave BFS some relational database-like quality similar to what we may finally see via WinFS in Windows 8. The 64-bit address space gave BFS the theoretical ability to support volumes of more than eight exobytes and files over 30 GB. This at a time when 30 GB hard drives were hardly commonplace.

Coupled with BeOS's performance-focused multithreaded core, BFS could provide high-performance streaming read, write and query access to storage with the ability to recover quickly after a failure. This made BeOS well-suited for audio and video manipulation, a task that it still accomplishes today in high-end media production systems.

There's a lot more to understand about the technical details of BFS. If you're curious to know more, take a look at Andrew Hudson's article at *Ars Technica* titled "The BeOS file system: an OS geek retrospective," along with a great interview with BFS creators Benoit Schillings and Dominic Giampaolo at *The Register*.

BeOS faded away as a commercial OS, but there's still a small, loyal group of enthusiasts keeping the flame burning. If you can find a BeOS 5 CD, it'll probably run on most commodity x86 hardware. Software is available from the BeBits repository. In addition, the Haiku project is an ongoing community effort to build a source-compatible open-source version of BeOS. They recently dropped an Alpha 2 release that's reasonably stable and runs most of the available legacy code.

IRIX, SGI Dogfight

Back in the early '90s, my employer struck a deal with Silicon Graphics to port our software to IRIX. I recall some Indy and Indigo boxes arriving at the office and a lot of oohing and ahhing among the staff. I don't recall whether we actually completed the ports -- probably not, given the state of things in the office. But as the eager young kid in the office I was given the exciting job of helping to set up the machines, which for the most part meant loading up applications from various tapes.

While waiting for the tapes to load up and spill their data, I did have a chance to explore the system. And one of the items I discovered hidden in the demos was a little gem called Dogfight. This little app was a 3D flight simulator that featured IP multicast-based multiplayer air combat over our humble little Ethernet network. Sure, it had frame rates and polygon counts you'd laugh at today, but at the time we'd never seen anything like it.

The first components of what would become the Dogfight demo were created by Gary Tarolli in the early '80s. OK, technically Dogfight wasn't an OS feature like some of the other items we've discussed here, but it was designed specifically to highlight the advanced (for the time) 3D rendering capabilities of SGI's systems. Building on his experience with IRIX and Dogfight, Tarolli went on to co-found 3dfx, which produced the Voodoo 3D graphics cards and Glide API -- used by some ground-breaking 3D PC games.

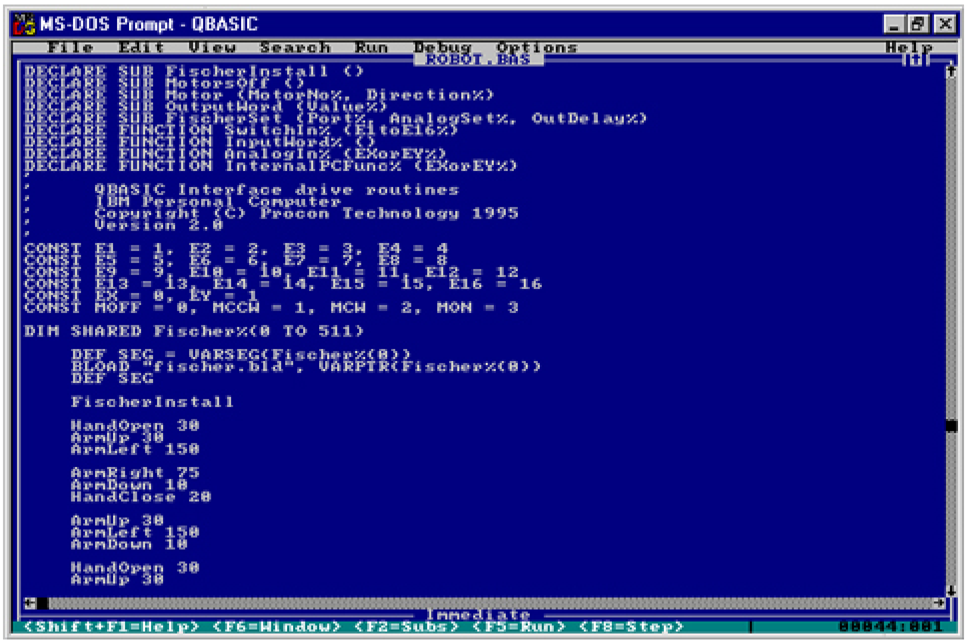
NeXTSTEP, Right-Click Context Menu

According to Wikipedia, the right-click "popup" menu technically originated in Xerox Alto's smalltalk environment. (We've all had a chance to play with one of those, right?) But the first time I experienced the right-click menu was when exploring NeXTSTEP on a friend's then-new NeXTstation. That makes some sense as there's an easily drawn line of inspiration from Alto to Macintosh to NeXT. While Mac OS did not embrace the right-click context menu until much later, it was an OS feature from the start in NeXTSTEP.

Of course, the context menu has become an intrinsic part of Windows, to the extent that it's possible to use right-clicking on pretty much anything in Windows or a Windows-based application as a discoverability tool. No need to look it up, just right-click and see what the options are. You could say the context menus provide useful task hints and shortcuts not unlike Tab command completion in a terminal or IDE.

That said, there are problems with the right-click context menu as currently implemented. First, the Windows context menu is getting rather unwieldy as it fills up with useful -- and esoteric -- options. Install a few applications that have shell integration and you can end up with a menu that contains nearly 20 items. On top of that, the APIs for shell integration make customizing the context menu difficult for anyone but programmers and superusers.

MS-DOS, BASIC


MS-DOS Prompt - QBASIC
File Edit View Search Run Debug Options Help
ROBOT.BAS
Immediate
<Shift+F1=Help> <F6=Window> <F2=Subs> <F5=Run> <F8=Step>
00044:001

```
DECLARE SUB FischerInstall ()
DECLARE SUB MotorsOff ()
DECLARE SUB Motor (MotorNo%, Direction%)
DECLARE SUB OutputWord (Value%)
DECLARE SUB FischerSet (Port%, AnalogSet%, OutDelay%)
DECLARE FUNCTION SwitchIn% (E1toE16%)
DECLARE FUNCTION InputWord% ()
DECLARE FUNCTION AnalogIn% (EXorEY%)
DECLARE FUNCTION InternalPCFunc% (EXorEY%)
'
'      QBASIC Interface drive routines
'      IBM Personal Computer
'      Copyright (C) Procon Technology 1995
'      Version 2.0
'
CONST E1 = 1, E2 = 2, E3 = 3, E4 = 4
CONST E5 = 5, E6 = 6, E7 = 7, E8 = 8
CONST E9 = 9, E10 = 10, E11 = 11, E12 = 12
CONST E13 = 13, E14 = 14, E15 = 15, E16 = 16
CONST EX = 0, EY = 1
CONST MOFF = 0, MCCW = 1, MCW = 2, MON = 3

DIM SHARED Fischer%(0 TO 511)

    DEF SEG = VARSEG(Fischer%(0))
    BLOAD "fischer.bld", VARPTR(Fischer%(0))
    DEF SEG

    FischerInstall

    HandOpen 30
    ArmUp 30
    ArmLeft 150

    ArmRight 75
    ArmDown 10
    HandClose 20

    ArmUp 30
    ArmLeft 150
    ArmDown 10

    HandOpen 30
    ArmUp 30
```

MS-DOS was undeniably the dominant desktop operating system throughout the '80s, and every one of those computers running MS-DOS included the Microsoft BASIC programming language in one form or another. In fact, the version of BASIC created by Paul Allen and Bill Gates predates even MS-DOS, originating as Altair BASIC in the '70s.

The BASIC language tools included in MS-DOS evolved over the years to include rudimentary Integrated Development Environment (IDE) features and a compiler for faster execution of programs. Microsoft BASIC, GW-BASIC, QuickBASIC and QBasic ultimately evolved into the Visual Basic language we know today, acquiring millions of enthusiasts along the way.

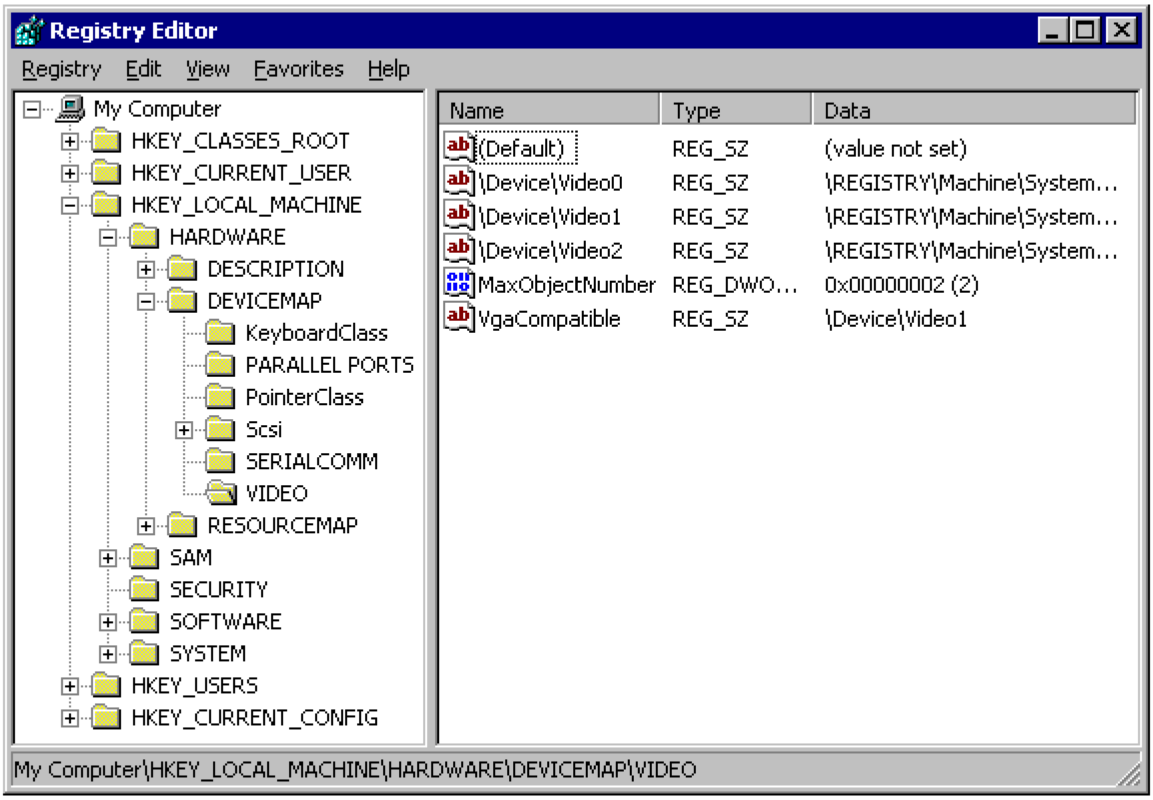
More recently, Microsoft DevLabs released an updated Windows interpretation of QBasic called Small Basic, which is intended to be used as a tool for teaching and experimentation.

What's significant about Microsoft BASIC is that it was shipped on tens of millions of computers -- in many cases, the first personal computers to make their way into offices and homes. It was the first opportunity to explore programming for a generation of computer users. Unless you worked in a technical occupation or studied computer science, your first exposure to programming most likely would have been through BASIC on MS-DOS. Hats off to Microsoft for democratizing the art of programming.

Windows Structure:

The registry is a hierarchical database that contains data that is critical for the operation of Windows and the applications and services that run on Windows. The data is structured in a tree format. Each node in the tree is called a *key*. Each key can contain both *subkeys* and data entries called *values*. Sometimes, the presence of a key is all the data that an application requires; other times, an application opens a key and uses the values associated with the key. A key can have any number of values, and the values can be in any form. For more information, see Registry Value Types and Registry Element Size Limits.

Each key has a name consisting of one or more printable characters. Key names are not case sensitive. Key names cannot include the backslash character (\), but any other printable character can be used. Value names and data can include the backslash character.
The name of each subkey is unique with respect to the key that is immediately above it in the hierarchy. Key names are not localized into other languages, although values may be.
The following illustration is an example registry key structure as displayed by the Registry Editor.



```
My Computer
  HKEY_CLASSES_ROOT
  HKEY_CURRENT_USER
  HKEY_LOCAL_MACHINE
    HARDWARE
      DESCRIPTION
      DEVICEMAP
        KeyboardClass
        PARALLEL PORTS
        PointerClass
        Scsi
        SERIALCOMM
        VIDEO
      RESOURCEMAP
    SAM
    SECURITY
    SOFTWARE
    SYSTEM
  HKEY_USERS
  HKEY_CURRENT_CONFIG
```

Name	Type	Data
(Default)	REG_SZ	(value not set)
\Device\Video0	REG_SZ	\REGISTRY\Machine\System...
\Device\Video1	REG_SZ	\REGISTRY\Machine\System...
\Device\Video2	REG_SZ	\REGISTRY\Machine\System...
MaxObjectNumber	REG_DWO...	0x00000002 (2)
VgaCompatible	REG_SZ	\Device\Video1

My Computer\HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\VIDEO

Each of the trees under **My Computer** is a key. The **HKEY_LOCAL_MACHINE** key has the following subkeys: **HARDWARE**,**SAM**, **SECURITY**, **SOFTWARE**, and **SYSTEM**. Each of these keys in turn has subkeys. For example, the **HARDWARE** key has the subkeys **DESCRIPTION**, **DEVICEMAP**, and **RESOURCEMAP**; the **DEVICEMAP** key has several subkeys including **VIDEO**.
Each value consists of a value name and its associated data, if any. **MaxObjectNumber** and **VgaCompatible** are values that contain data under the **VIDEO** subkey.
A registry tree can be 512 levels deep. You can create up to 32 levels at a time through a single registry API call.

Recycle Bin:

Similar to the Apple's Trash application, the Recycle Bin is a location where deleted files or folders are temporarily stored in every version of Microsoft Windows since Windows 95. The

Recycling Bin allows users to recover files that have been deleted in Windows and can be found on the desktop. The image to the right is an example of what the Recycle Bin may look like in your version of Windows.

Note: In newer versions of Windows the Recycle Bin icon may be hidden by default. If this icon is missing, you may need to show the Recycle Bin to see it on the desktop.

Tip: You can hold down the Shift key when deleting a file to prevent it from going to the Recycle Bin.

Note: If **you delete a file in the Windows command line, it is not sent to the Recycle Bin.**

- Adding an item to the Recycle Bin
- How to open or find what is in the Recycle Bin
- Why is my Recycle Bin icon missing?
- Why is the Recycle Bin emptied?
- How long do items stay in the Recycle Bin?
- Is the Recycle Bin a software program?
- Related Recycle Bin pages.

Adding an item to the Recycle Bin

When you delete a file or folder, that file or folder is added, or sent, to the Recycle Bin.

- How to delete a file, directory, or folder.

How to open or find what is in the Recycle Bin

To open the Recycle Bin, double-click on the Recycle Bin icon on the computer desktop. If done correctly a new Recycle Bin window will open showing all of the files in the Recycle Bin.

Tip: If the Recycle Bin icon is not found on your computer Desktop, see the next section below.

Linux

Linux

- Linux basics
- Basic Linux Commands

Linux Basics:

Linux is a free, open-source operating system. All of DigitalOcean's offered operating systems are Linux distributions. Linux has been under active development since 1991. It has evolved to be versatile and is used all over the world, from web servers to cellphones.DigitalOcean offers Linux distributions on droplets because Linux is free and easy to use. However, newcomers to Linux may find it difficult to approach the structure of an unfamiliar operating system. This guide gently introduces key terminal skills and equips newcomers to learn more about Linux.

The Terminal:

For most of the time you access a cloud server, you'll be doing it through a terminal shell. The shell allows you to execute commands on the droplet.

All administrative tasks can be accomplished through the terminal. This includes file manipulation, package installation, and user management.

The terminal is interactive. You specify commands to run. The terminal outputs the results of those commands. Executing any command is done by typing it and pressing Enter.

Navigation:

Linux filesystems are based on a directory tree. This means that you can create directories (or "folders") inside other directories, and files can exist in any directory.

Basic Linux Commands:

1. ls:

The ls command - the list command - functions in the Linux terminal to show all of the major directories filed under a given file system. For example, the command:

ls /applications will show the user all of the folders stored in the overall applications folder.The ls command is used for viewing files, folders and directories.

2. cd:

The cd command - change directory - will allow the user to change between file directories. As the name command name suggest, you would use the cd command to circulate between two different directories. For example, if you wanted to change from the home directory to the Arora directory, you would input the following command:

cd/arora/applications

As you might have noted, the path name listed lists in reverse order. Logically **cd/arora/applications** reads change to the arora directory which is stored in the applications directory. All Linux commands follow a logical path.

3. mv:
The mv command - move - allows a user to move a file to another folder or directory. Just like dragging a file located on a PC desktop to a folder stored within the "Documents" folder, the mv command functions in the same manner. An example of the mv command is:

```
mv/arora/applications/majorapps /arora/applications/minorapps
```

The first part of the command **mv/arora/applications/majorapps** lists the application to be moved. In this case, arora. The second part of the command **/arora/applications/minorapps** lists where arora will be moved to - from majorapps to minorapps.

4. man:
The man command - the manual command - is used to show the manual of the inputted command. Just like a film on the nature of film, the man command is the meta command of the Linux CLI. Inputting the man command will show you all information about the command you are using. An example: **man cd**

The inputting command will show the manual or all relevant information for the change directory command.

5. mkdir:
The mkdir - make directory - command allows the user to make a new directory. Just like making a new directory within a PC or Mac desktop environment, the mkdir command makes new directories in a Linux environment. An example of the mkdir command

mkdir testdirectory

The example command made the directory "testdirectory".

6. rmdir:
The rmdir - remove directory - command allows the user to remove an existing command using the Linux CLI. An example of the rmdir command:

rmdir testdirectory

The example command removed the directory "testdirectory".

It should be noted: both the mkdir and rmdir commands make and remove directories. They do not make files and they will also not remove a directory which has files in it. The mkdir will make an empty directory and the rmdir command will remove an empty directory**.**

7. touch:
The touch command - a.k.a. the make file command - allows users to make files using the Linux CLI. Just as the mkdir command makes directories, the touch command makes files.

Just as you would make a .doc or a .txt using a PC desktop, the touch command makes empty files. An example of the touch command: **touch testfile.txt**

The example touch command effectively created the file testfile.txt. As noted by the extension, the file created is a .txt or text file. To equate, a .txt file in Linux is akin to a .txt notebook file within a Windows or Mac OS.

8. rm:

The rm command - remove - like the rmdir command is meant to remove files from your Linux OS. Whereas the rmdir command will remove directories and files held within, the rm command will delete created files. An example of the rm command:

rm testfile.txt

The aforementioned command removed testfile.txt. Interestingly, whereas the rmdir command will only delete an empty directory, the rm command will remove both files and directories with files in it. This said, the rm command carries more weight than the rmdir command and should be used with more specificity.

9. locate:

The locate - a.k.a. find - command is meant to find a file within the Linux OS. If you don't know the name of a certain file or you aren't sure where the file is saved and stored, the locate command comes in handy. A locate command example:

locate -i *red*housecity***

The stated command will locate an file with the a file name containing "Red", "House" and "City". A note on the input: the use of "-i" tells the system to search for a file unspecific of capitalization (Linux functions in lower case). The use of the asterik "*" signifies searching for a wildcard. A wildcard tells the system to pull any and all files containing the search criteria.

By specifying -i with wildcards, the locate CLI command will pull back all files containing your search criteria effectivley casting the widest search net the system will allow.

10. clear

The clear command does exactly what it says. When your Linux CLI gets all mucked up with various readouts and information, the clear command clears the screen and wipes the board clean. Using the clear command will take the user back to the start prompt of whatever directory you are currently operating in. To use the clear command simply type **clear**.

Microsoft Office (MS Word, Excel, Power Point)

- **Word Processing: MS Word**
- **Worksheet: MS Excel**
- **Presentation Graphics: MS Power Point**

MS Word:

Microsoft Word (or simply Word) is a word processor developed by Microsoft. It was first released on October 25, 1983 under the name Multi-Tool
Word for Xenix systems. Subsequent versions were later written for several other platforms including IBM PCs running DOS (1983), Apple Macintosh running the Classic Mac OS(1985), AT&T Unix PC (1985), Atari ST (1988), OS/2 (1989), Microsoft
Windows (1989), SCO Unix (1994), and OS X (2001). Commercial versions of Word are licensed as a standalone product or as a component of Microsoft Office, Windows RT or the discontinued Microsoft Works suite. Microsoft Word Viewer and Office
Online are freeware editions of Word with limited features.

In 1981, Microsoft hired Charles Simonyi, the primary developer of Bravo, the first GUI word processor, which was developed at Xerox PARC. Simonyi started work on a word processor called Multi-Tool *Word* and soon hired Richard Brodie, a former Xerox intern, who became the primary software engineer.

Microsoft announced Multi-Tool Word for Xenix and MS-DOS in 1983. Its name was soon simplified to *Microsoft* Word. Free demonstration copies of the application were bundled with the November 1983 issue of *PC World*, making it the first to be distributed on-disk with a magazine. That year Microsoft demonstrated Word running on Windows.

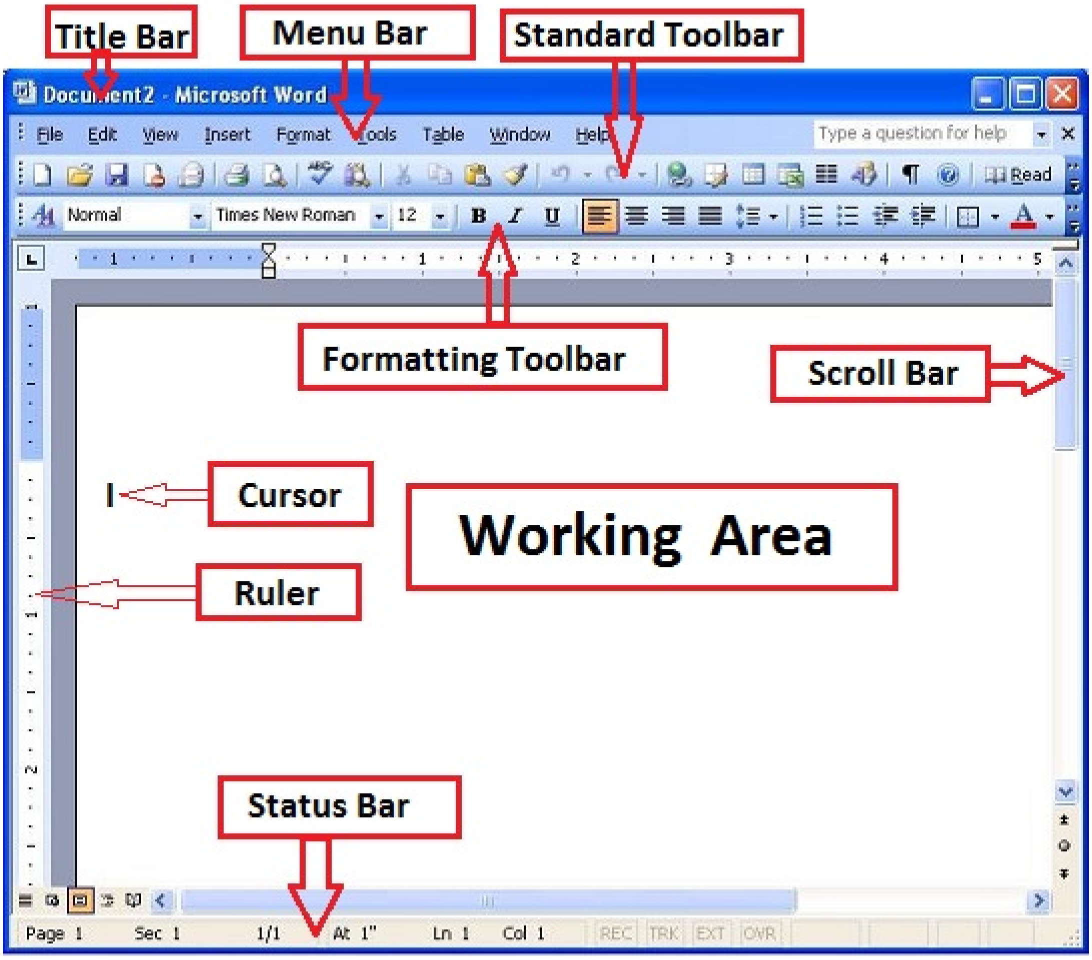
Unlike most MS-DOS programs at the time, Microsoft Word was designed to be used with a mouse.[11]Advertisements depicted the Microsoft Mouse, and described Word as a WYSIWYG, windowed word processor with the ability to undo and display bold, italic, and underlined text, although it could not render fonts. It was not initially popular, since its user interface was different from the leading word processor at the time, WordStar.[15] However, Microsoft steadily improved the product, releasing versions 2.0 through 5.0 over the next six years. In 1985, Microsoft ported Word to the classic Mac OS (known as Macintosh System Software at the time). This was made easier by Word for DOS having been designed for use with high-resolution displays and laser printers, even though none were yet available to the general public. Following the precedents of LisaWrite and MacWrite, Word for Mac OS added true WYSIWYG features. It fulfilled a need for a word processor that was more capable than MacWrite After its release, Word for Mac OS's sales were higher than its MS-DOS counterpart for at least four years.[8]

The second release of Word for Mac OS, shipped in 1987, was named Word 3.0 to synchronize its version number with Word for DOS; this was Microsoft's first attempt to synchronize version numbers across platforms. Word 3.0 included numerous internal enhancements and new features, including the first implementation of the Rich Text Format (RTF) specification, but was plagued with bugs. Within a few months, Word 3.0 was superseded by a more stable Word 3.01, which was mailed free to all registered users of 3.0. After MacWrite Pro was discontinued in the mid-1990s, Word for Mac OS never had any serious rivals. Word 5.1 for Mac OS, released in 1992, was a very popular word processor owing to its elegance, relative ease of use and feature set. Many users say it is the best version of Word for Mac OS ever created.

In 1986, an agreement between Atari and Microsoft brought Word to the Atari under the name *Microsoft Write*. The Atari ST version was a port of Word 1.05 for the Mac OS and was never updated due to the outstanding degree of software piracy on the Atari platform.

The first version of Word for Windows was released in 1989. With the release of Windows 3.0 the following year, sales began to pick up and Microsoft soon became the market leader for word processors for IBM PC-compatible computers. In 1991, Microsoft capitalized on Word for Windows' increasing popularity by releasing a version of Word for DOS, version 5.5, that replaced its unique user interface with an interface similar to a Windows application. When Microsoft became aware of the Year 2000 problem, it made Microsoft Word 5.5 for DOS available for download free. As of March 2014, it is still available for download from Microsoft's web site.In 1991, Microsoft embarked on a project code-named Pyramid to completely rewrite Microsoft Word from the ground up. Both the Windows and Mac OS versions would start from the same code base. It was abandoned when it was determined that it would take the development team too long to rewrite and then catch up with all the new capabilities that could have been added in the same time without a rewrite. Instead, the next versions of Word for Windows and Mac OS, dubbed version 6.0, both started from the code base of Word for Windows 2.0.

With the release of Word 6.0 in 1993, Microsoft again attempted to synchronize the version numbers and coordinate product naming across platforms, this time across DOS, Mac OS, and Windows (this was the last version of Word for DOS). It introduced AutoCorrect, which automatically fixed certain typing errors, and AutoFormat, which could reformat many parts of a document at once. While the Windows version received favorable reviews (e.g., from *InfoWorld*), the Mac OS version was widely derided. Many accused it of being slow, clumsy and memory intensive, and its user interface differed significantly from Word 5.1. In response to user requests, Microsoft offered Word 5 again, after it had been discontinued. Subsequent versions of Word for macOS are no longer direct ports of Word for Windows, instead featuring a mixture of ported code and native code.



File extensions:

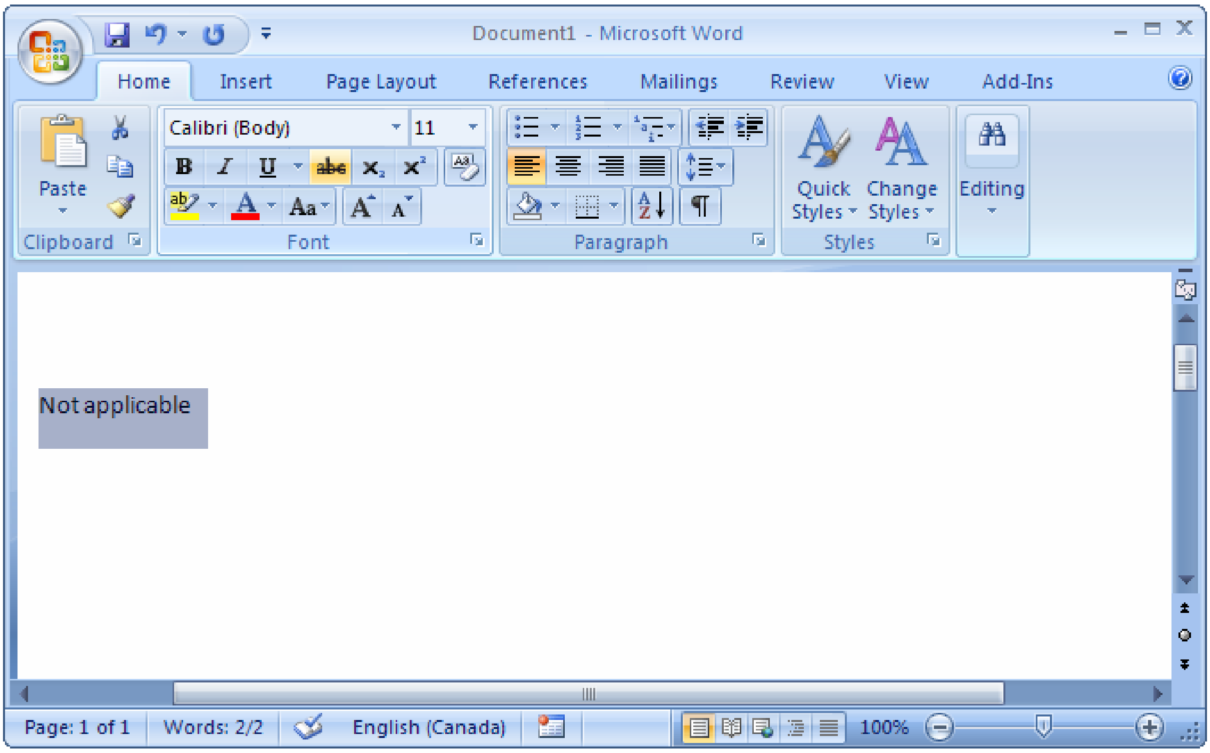
Microsoft Word's native file formats are denoted either by a .doc or .docx filename extension.

Although the .doc extension has been used in many different versions of Word, it actually encompasses four distinct file formats:

1. Word for DOS
2. Word for Windows 1 and 2; Word 3 and 4 for Mac OS
3. Word 6 and Word 95 for Windows; Word 6 for Mac OS
4. Word 97 and later for Windows; Word 98 and later for Mac OS

The newer .docx extension signifies the Office Open XML international standard for Office documents and is used by Word 2007 and later for Windows, Word 2008 and later for macOS, as well as by a growing number of applications from other vendors, including OpenOffice.org Writer, an open source word processing program.

Word for Windows:



Not applicable

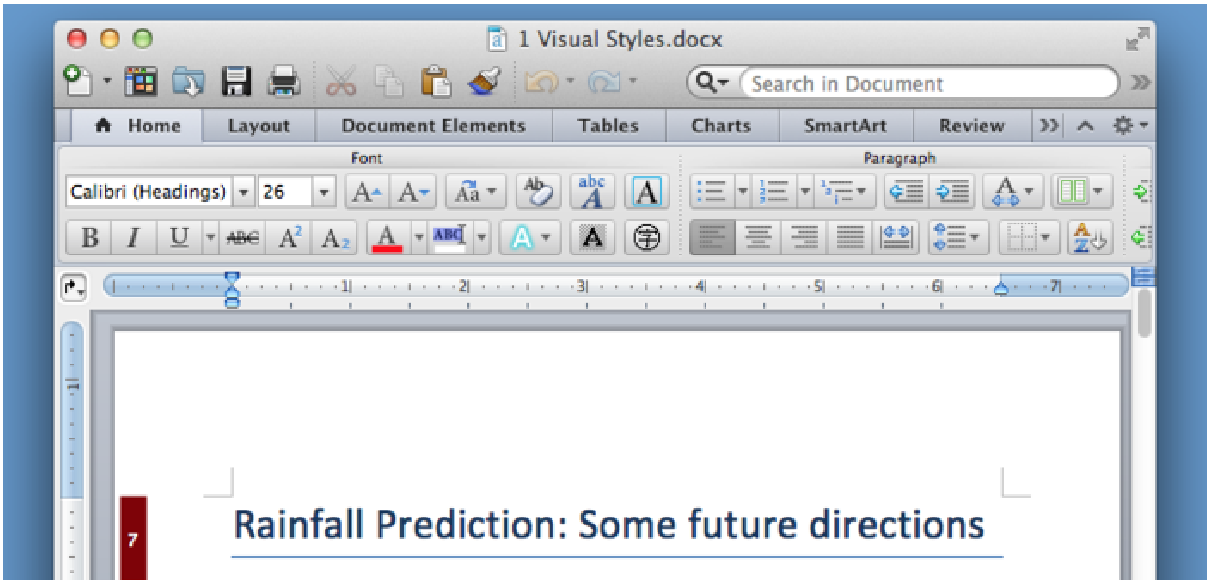
Microsoft Word 2007

Word for Windows is available stand-alone or as part of the Microsoft Office suite. Word contains rudimentary desktop publishing capabilities and is the most widely used word processing program on the market. Word files are commonly used as the format for sending text documents via e-mail because almost every user with a computer can read a Word document by using the Word application, a Word viewer or a word processor that imports the Word format (see Microsoft Word Viewer). Word 6 for Windows NT was the first 32-bit version of the product, released with Microsoft Office for Windows NT around the same time as Windows 95. It was a straightforward port of Word 6.0. Starting with Word 95, releases of Word were named after the year of its release, instead of its version number.

Word 2010 allows more customization of the Ribbon, adds a Backstage view for file management, has improved document navigation, allows creation and embedding of screenshots,[31] and integrates with Word Web App.

Word for Mac:

Microsoft Office § Mac versions



Rainfall Prediction: Some future directions

Microsoft Word 2011 running on OS X

In 1997, Microsoft formed the Macintosh Business Unit as an independent group within Microsoft focused on writing software for Mac OS. Its first version of Word, Word 98, was released with Office 98 Macintosh Edition. Document compatibility reached parity with Word 97, and it included features from Word 97 for Windows, including spell and grammar checking with squiggles. Users could choose the menus and keyboard shortcuts to be similar to either Word 97 for Windows or Word 5 for Mac OS.

Word 2001, released in 2000, added a few new features, including the Office Clipboard, which allowed users to copy and paste multiple items. It was the last version to run on classic Mac OS and, on Mac OS X, it could only run within the Classic Environment. Word X, released in 2001, was the first version to run natively on, and required, Mac OS X and introduced non-contiguous text selection.

Word 2004 was released in May 2004. It included a new Notebook Layout view for taking notes either by typing or by voice.Other features, such as tracking changes, were made more similar with Office for Windows.

Word 2008, released on January 15, 2008, included a Ribbon-like feature, called the Elements Gallery, that can be used to select page layouts and insert custom diagrams and images. It also included a new view focused on publishing layout, integrated bibliography management, and native support for the new Office Open XML format. It was the first version to run natively on Intel-based Macs.Word 2011, released in October 2010, replaced the Elements Gallery in favor of a Ribbon user interface that is much more similar to Office for Windows, and includes a full-screen mode that allows users to focus on reading and writing documents, and support for.

MS – Excel:

Microsoft Excel is a spreadsheet developed by Microsoft for by Windows, by Microsoft Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. It has been a very widely applied spreadsheet for these platforms, especially since version 5 in 1993, and it has replaced Lotus 1-2-3 as the industry standard for spreadsheets. Excel forms part of Microsoft Office.

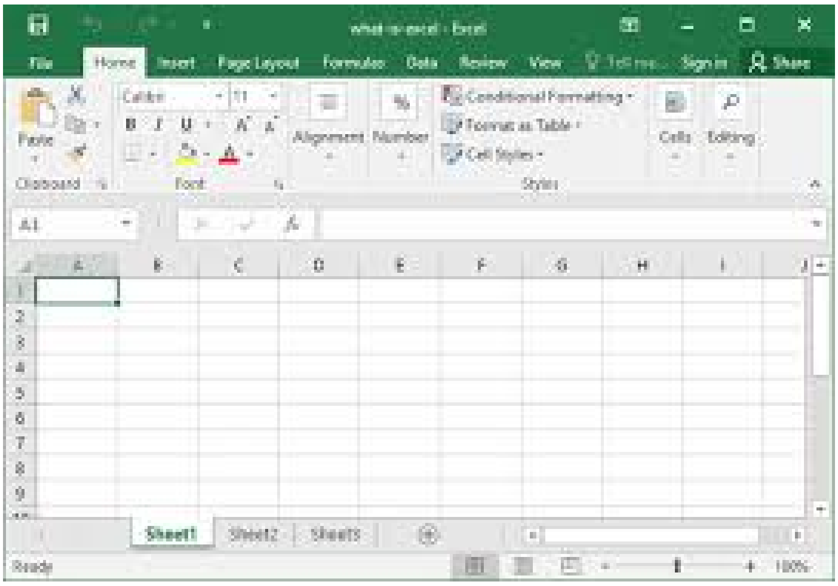
A **spreadsheet** is an interactive computer application for organization, analysis and storage of data in tabular form. Spreadsheets are developed as computerized simulations of paper accounting worksheets.[4] The program operates on data entered in cells of a table. Each cell may contain either numeric or text data, or the results of formulas that automatically calculate and display a value based on the contents of other cells. A spreadsheet may also refer to one such electronic document.[5][6][7]

Spreadsheet users can adjust any stored value and observe the effects on calculated values. This makes the spreadsheet useful for "what-if" analysis since many cases can be rapidly investigated without manual recalculation. Modern spreadsheet software can have multiple interacting sheets, and can display data either as text and numerals, or in graphical form.

Besides performing basic arithmetic and mathematical functions, modern spreadsheets provide built-in functions for common financial and statistical operations. Such calculations as net present value or standard deviation can be applied to tabular data with a pre-programmed function in a formula. Spreadsheet programs also provide conditional expressions, functions to convert between text and numbers, and functions that operate on strings of text.

Spreadsheets have replaced paper-based systems throughout the business world. Although they were first developed for accounting or bookkeeping tasks, they now are used extensively in any context where tabular lists are built, sorted, and shared.

LANPAR, available in 1969 was the first electronic spreadsheet on mainframe and time sharing computers. LANPAR was an acronym: LANguage for Programming Arrays at Random. VisiCalc was the first electronic spreadsheet on a microcomputer,[9] and it helped turn the Apple II computer into a popular and widely used system. Lotus 1-2-3 was the leading spreadsheet when DOS was the dominant operating system. Excel now has the largest market share on the Windows and Macintosh platforms. A spreadsheet program is a standard feature of an office productivity suite; since the advent of web apps, office suites now also exist in web app form. Web based spreadsheets, such as Microsoft Excel Online, Google Sheets, Yahoo Sheets and Zoho Sheets, are a relatively new category.



A **spreadsheet** consists of a table of *cells* arranged into rows and columns and referred to by the X and Y locations. X locations, the columns, are normally represented by letters, "A", "B", "C", etc., while rows are normally represented by numbers, 1, 2, 3, etc. A single cell can be referred to by addressing its row and column, "C10" for instance. This electronic concept of cell references was first introduced in LANPAR (Language for Programming Arrays at Random) (co-invented by Rene Pardo and Remy Landau) and a variant used in VisiCalc, and known as "A1 notation". Additionally, spreadsheets have the concept of a *range*, a group of cells, normally contiguous. For instance, one can refer to the first ten cells in the first column with the range "A1:A10". LANPAR innovated forward referencing/natural order calculation which didn't re-appear until Lotus 123 and Microsoft's MultiPlan Version 2.

In modern spreadsheet applications, several spreadsheets, often known as *worksheets* or simply *sheets*, are gathered together to form a *workbook*. A workbook is physically represented by a file, containing all the data for the book, the sheets and the cells with the sheets. Worksheets are normally represented by tabs that flip between pages, each one containing one of the sheets, although Numbers changes this model significantly. Cells in a multi-sheet book add the sheet name to their reference, for instance, "Sheet 1!C10". Some systems extend this syntax to allow cell references to different workbooks.

Users interact with sheets primarily through the cells. A given cell can hold data by simply entering it in, or a formula, which is normally created by preceding the text with an equals sign. Data might include the string of text hello world, the number 5 or the date 16-Dec-91. A formula would begin with the equals sign, =5*3, but this would normally be invisible because the display shows the *result* of the calculation, 15 in this case, not the formula itself. This may lead to confusion in some cases.

The key feature of spreadsheets is the ability for a formula to refer to the contents of other cells, which may in turn be the result of a formula. To make such a formula, one simply replaces a number with a cell reference. For instance, the formula =5*C10 would produce the result of multiplying the value in cell C10 by the number 5. If C10 holds the value 3 the result will be 15. But C10 might also hold its own formula referring to other cells, and so on.

The ability to chain formulas together is what gives a spreadsheet its power. Many problems can be broken down into a series of individual mathematical steps, and these can be assigned to individual formulas in cells. Some of these formulas can apply to ranges as well, like the SUM function that adds up all the numbers within a range.

Spreadsheets share many principles and traits of databases, but spreadsheets and databases are not the same thing. A spreadsheet is essentially just one table, whereas a database is a collection of many tables with machine-readable semantic relationships between them. While it is true that a workbook that contains three sheets is indeed a file containing multiple tables that can interact with each other, it lacks the relational structure of a database. Spreadsheets and databases are interoperable—sheets can be imported into databases to become tables within them, and database queries can be exported into spreadsheets for further analysis.

A spreadsheet program is one of the main components of an office productivity suite, which usually also contains a word processor, a presentation program, and a database management system. Programs within a suite use similar commands for similar functions. Usually sharing data between the components is easier than with a non-integrated collection of functionally equivalent programs. This was particularly an advantage at a time when many personal computer systems used text-mode displays and commands, instead of a graphical user interface.

Paper spreadsheets:

The word "spreadsheet" came from "spread" in its sense of a newspaper or magazine item (text or graphics) that covers two facing pages, extending across the center fold and treating the two pages as one large one. The compound word "spread-sheet" came to mean the format used to present book-keeping ledgers—with columns for categories of expenditures across the top, invoices listed down the left margin, and the amount of each payment in the cell where its row and column intersect—which were, traditionally, a "spread" across facing pages of a bound ledger (book for keeping accounting records) or on oversized sheets of paper (termed "analysis paper") ruled into rows and columns in that format and approximately twice as wide as ordinary paper.

Batch spreadsheet report generator:

A batch "spreadsheet" is indistinguishable from a batch compiler with added input data, producing an output report, i.e., a 4GL or conventional, non-interactive, batch computer program. However, this concept of an electronic spreadsheet was outlined in the 1961 paper "Budgeting Models and System Simulation" by Richard Mattessich.[15] The subsequent work by Mattessich (1964a, Chpt. 9, Accounting and Analytical Methods) and its companion volume, Mattessich (1964b, Simulation of the Firm through a Budget Computer Program) applied computerized spreadsheets to accounting and budgeting systems (on mainframe computers programmed in FORTRAN IV). These batch Spreadsheets dealt primarily with the addition or subtraction of entire columns or rows (of input variables), rather than individual cells.

MS Power Point:

Microsoft PowerPoint (or simply **PowerPoint**) is a presentation program,[4] created by Robert Gaskins and Dennis Austin[4] at a software company named Forethought, Inc.[4] It was released on April 20, 1987,[5] initially for Macintosh computers only.[4] Microsoft acquired PowerPoint for $14 million three months after it appeared.[6] This was Microsoft's first significant acquisition, and Microsoft set up a new business unit for PowerPoint in Silicon Valley where Forethought had been located.]Microsoft PowerPoint is one of many programs run by the company Microsoft and can be identified by its trademark orange, and P initial on the logo. It offers users many ways to display information from simple presentations to complex multimedia presentations.

PowerPoint became a component of the Microsoft Office suite, first offered in 1989 for Macintosh[8] and in 1990 for Windows,[9] which bundled several Microsoft apps. Beginning with PowerPoint 4.0 (1994), PowerPoint was integrated into Microsoft Office development, and adopted shared common components and a converged user interface.

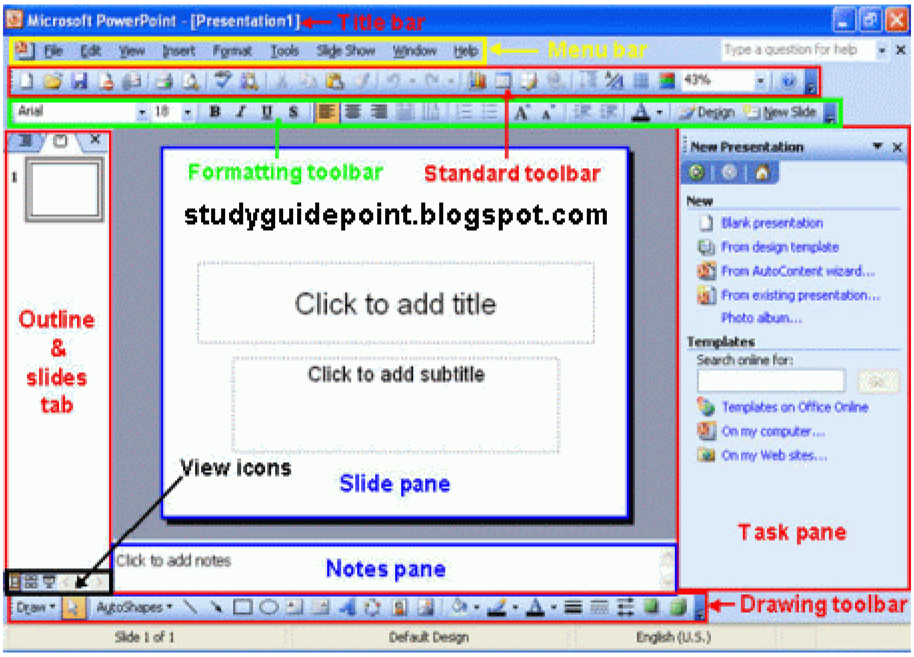
PowerPoint's market share was very small at first, prior to introducing a version for Microsoft Windows, but grew rapidly with the growth of Windows and of Office. Since the late 1990s, PowerPoint's worldwide market share of presentation software has been estimated at 95 percent.

PowerPoint was originally designed to provide visuals for group presentations within business organizations, but has come to be very widely used in many other communication situations, both in business and beyond. The impact of this much wider use of PowerPoint has been experienced as a powerful change throughout society, with strong reactions including advice that it should be used less, should be used differently, or should be used better.

The first PowerPoint version (Macintosh 1987) was used to produce overhead transparencies, the second (Macintosh 1988, Windows 1990) could also produce color 35mm slides. The third version (Windows and Macintosh 1992) introduced video output of virtual slideshows to digital projectors, which would over time completely replace physical transparencies and slides. A dozen major versions since then have added many additional features and modes of operation and have made PowerPoint available beyond Apple Macintosh and Microsoft Windows, adding versions for iOS, Android, and web access.

PowerPoint was created by Robert Gaskins and Dennis Austin at a software startup in Silicon Valley named Forethought, Inc.[20] Forethought had been founded in 1983 to create an integrated environment and applications for future personal computers that would provide a graphical user interface, but it had run into difficulties requiring a "restart" and new plan.

On July 5, 1984, Forethought hired Robert Gaskins as its vice president of product development to create a new application that would be especially suited to the new graphical personal computers, such as Microsoft Windows and Apple Macintosh.Gaskins produced his initial description of PowerPoint about a month later (August 14, 1984) in the form of a 2-page document titled "Presentation Graphics for Overhead Projection." By October 1984 Gaskins had selected Dennis Austin to be the developer for PowerPoint. Gaskins and Austin worked together on the definition and design of the new product for nearly a year, and produced the first specification document dated August 21, 1985.This first design document showed a product as it would look in Microsoft Windows 1.0, which at that time had not been released.



Development from that spec was begun by Austin in November 1985, for Macintosh first.About six months later, on May 1, 1986, Gaskins and Austin chose a second developer to join the project, Thomas Rudkin.Gaskins prepared two final product specification marketing documents in June 1986; these described a product for both Macintosh and Windows. At about the same time, Austin, Rudkin, and Gaskins produced a second and final major design specification document, this time showing a Macintosh look.

Throughout this development period the product was called "Presenter." Then, just before release, there was a last-minute check with Forethought's lawyers to register the name as a trademark, and "Presenter" was unexpectedly rejected because it had already been used by someone else. Gaskins says that he thought of "PowerPoint", based on the product's goal of "empowering" individual presenters, and sent that name to the lawyers for clearance, while all the documentation was hastily revised.

Funding to complete development of PowerPoint was assured in mid-January, 1987, when a new Apple Computer venture capital fund, called Apple's Strategic Investment Group,[selected PowerPoint to be its first investment. A month later, on February 22, 1987, Forethought announced PowerPoint at the Personal Computer Forum in Phoenix; John Sculley, the CEO of Apple, appeared at the announcement and said "We see desktop presentation as potentially a bigger market for Apple than desktop publishing."PowerPoint 1.0 for Macintosh shipped from manufacturing on April 20, 1987, and the first production run of 10,000 units was sold out.

Programming Language

C Language :

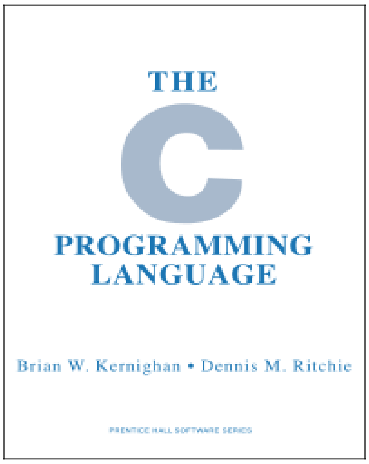
Introduction of C Language:

The C Programming Language (sometimes termed ***K&R***, after its authors' initials) is a computer programming book written by Brian Kernighan and Dennis Ritchie, the latter of whom originally designed and implemented the language, as well as co-designed the Unix operating system with which development of the language was closely intertwined. The book was central to the development and popularization of the C programming language and is still widely read and used today. Because the book was co-authored by the original language designer, and because the first edition of the book served for many years as the de facto standard for the language, the book was regarded by many to be the authoritative reference on C.

The first edition of the book, published in 1978, was the first widely available book on the C programming language. C was created by Dennis Ritchie. Brian Kernighan wrote the first C tutorial.[3] The authors came together to write the book in conjunction with the language's early development at AT&T Bell Labs. The version of C described in this book is sometimes termed *K&R C*(after the book's authors), often to distinguish this early version from the later version of C standardized as *ANSI C*.

In 1988, the second edition of the book was published, updated to cover the changes to the language resulting from the then-new ANSI C standard, particularly with the inclusion of reference material on standard libraries. The second edition (and as of 2018, the most recent edition) of the book has since been translated into over 20 languages. In 2012, an eBook version of the second edition was published in ePub, Mobi, and PDF formats.

ANSI C, first standardized in 1989 (as ANSI X3.159-1989), has since undergone several revisions, the most recent of which is ISO/IEC 9899:2011 (also termed *C11*), adopted as an ANSI standard in October 2011. However, no new edition of *The C Programming Language* has been issued to cover the more recent standards.


THE
C
PROGRAMMING
LANGUAGE
Brian W. Kernighan • Dennis M. Ritchie
PRENTICE HALL SOFTWARE SERIES

Character set of C

character:- It denotes any alphabet, digit or special symbol used to represent information.

Use:-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.

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 character set, 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.

Variables and Key words:

Variables and Keywords in C:- A **variable** in simple terms is a storage place which has some memory allocated to it. So basically a variable used to store some form of data. Different types of variables require different amounts of memory and have some specific set of operations which can be applied on them.

Variable Declaration:
A typical variable declaration is of the form:

```
type variable_name;

  or for multiple variables:

type variable1_name, variable2_name, variable3_name;
```

A variable name can consist of alphabets (both upper and lower case), numbers and the underscore '_' character. However, the name must not start with a number.

Difference b/w variable declaration and definition:
Variable declaration refers to the part where a variable is first declared or introduced before its first use. Variable definition is the part where the variable is assigned a memory location and a value. Most of the times, variable declaration and definition are done together
See the following C program for better clarification:

```
#include <stdio.h>
int main()
{
   // declaration and definition of variable 'a123'
   char a123 = 'a';

   // This is also both declaration and definition as 'b' is allocated
   // memory and assigned some garbage value.
   float b;

   // multiple declarations and definitions
   int _c, _d45, e;

   // Let us print a variable
   printf("%c \n", a123);

   return 0;
}
```

OUTPUT: a

Keywords are specific reserved words in C each of which has a specific feature associated with it. Almost all of the words which help us use the functionality of the C language are included in the list of keywords. So you can imagine that the list of keywords is not going to be a small one!
There are a total of 32 keywords in C:

```
auto     break   case    char    const    continue
default  do      double  else    enum     extern
float    for     goto    if      int      long
register return  short   signed  sizeof   static
struct   switch  typedef union   unsigned void
volatile while
```

Most of these keywords have already been discussed in the various sub-sections of the C language, like Data Types, Storage Classes, Control Statements, Functions etc.
Let us discuss some of the other keywords which allow us to use the basic functionality of C:

const: const can be used to declare constant variables. Constant variables are variables which when initialized, can't change their value. Or in other words, the value assigned to them is unmodifiable.
Syntax:

```
const data_type var_name = var_value;
```

Note: Constant variables need to be compulsorily be initialized during their declaration itself. const keyword is also used with pointers. Please refer the const qualifier in C for understanding the same.

extern: extern simply tells us that the variable is defined elsewhere and not within the same block where it is used. Basically, the value is assigned to it in a different block and this can be overwritten/changed in a different block as well. So an extern variable is nothing but a global variable initialized with a legal value where it is declared in order to be used elsewhere. It can be accessed within any function/block. Also, a normal global variable can me made extern as well by placing the 'extern' keyword before its declaration/definition in any function/block. This basically signifies that we are not initializing a new variable but instead we are using/accessing the global variable only. The main purpose of using extern variables is that they can be accessed between two different files which are part of a large program.
Syntax:

```
extern data_type var_name = var_value;
```

static: static keyword is used to declare static variables which are popularly used while writing programs in C language. Static variables have a property of preserving their value even after they are out of their scope! Hence, static variables preserve the value of their last use in their scope. So we can say that they are initialized only once and exist till the termination of the program. Thus, no new memory is allocated because they are not re-declared. Their scope is local to the function to which they were defined. Global static variables can be accessed anywhere in the program. By default, they are assigned the value 0 by the compiler.
Syntax:

```
static data_type var_name = var_value;
```

void: void is a special data type only. But what makes it so special? void, as it literally means an empty data type. It means it has nothing or it holds no value. For example, when it is used as the return data type for a function, it simply represents that the function returns no value. Similarly, when it is added to a function heading, it represents that the function takes no arguments.

Note: void also has a significant use with pointers. Please refer the void pointer in C for understanding the same.

typedef: typedef is used to give a new name to an already existing or even a custom data type (like a structure). It comes in very handy at times, for example in a case when the name of the structure defined by you is very long or you just need a short-hand notation of a per-existing data type.

Let's implement the keywords which we have discussed above. See the following code which is a working example to demonstrate these keywords:

```
#include <stdio.h>

 // declaring and initializing an extern variable

extern int x = 9;

 // declaring and initializing a global variable

// simply int z; would have initialized z with

// the default value of a global variable which is 0

int z=10;

 // using typedef to give a short name to long long int

// very convenient to use now due to the short name

typedef long long int LL;

 // function which prints square of a no. and which has void as its

// return data type

void calSquare(int arg)

{

   printf("The square of %d is %d\n",arg,arg*arg);

}

 // Here void means function main takes no parameters

int main(void)
```

```
{
    // declaring a constant variable, its value cannot be modified
    const int a = 32;
 // declaring a  char variable
    char b = 'G';
  // telling the compiler that the variable z is an extern variable
    // and has been defined elsewhere (above the main function)
 extern int z;
LL c = 1000000;
printf("Hello World!\n");
  // printing the above variables
 printf("This is the value of the constant variable 'a': %d\n",a);
 printf("'b' is a char variable. Its value is %c\n",b);
 printf("'c' is a long long int variable. Its value is %lld\n",c);
 printf("These are the values of the extern variables 'x' and 'z'"
    " respectively: %d and %d\n",x,z);
 // value of extern variable x modified
    x=2;
 // value of extern variable z modified
    z=5;
 // printing the modified values of extern variables 'x' and 'z'
    printf("These are the modified values of the extern variables"
    " 'x' and 'z' respectively: %d and %d\n",x,z);
 // using a static variable
    printf("The value of static variable 'y' is NOT initialized to 5 after the "
            "first iteration! See for yourself :)\n");
 while (x > 0)
    {
        static int y = 5;
        y++;
        // printing value at each iteration
        printf("The value of y is %d\n",y);
        x--;
}
 // print square of 5
    calSquare(5);
  printf("Bye! See you soon. :)\n");
 return 0;
}
```

Output:

```
Hello World
This is the value of the constant variable 'a': 32
'b' is a char variable. Its value is G
'c' is a long long int variable. Its value is 1000000
These are the values of the extern variables 'x' and 'z' respectively: 9 and 10
These are the modified values of the extern variables 'x' and 'z' respectively: 2 and 5
The value of static variable 'y' is NOT initialized to 5 after the first iteration! See for yourself
:)
The value of y is 6
The value of y is 7
The square of 5 is 25
Bye! See you soon. :)
```

Decision and Control Structure:

- **if ,if-elses, forms of if-elses statement**
- **Operator**
- **Types of Loops**
- **Case and Switch**

1)IF:- An **if** statement consists of a Boolean expression followed by one or more statements.

Syntax

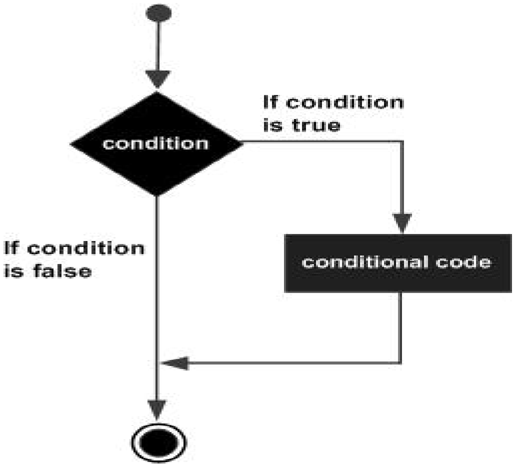
The syntax of an 'if' statement in C programming language is −

```
if(boolean_expression) {
  /* statement(s) will execute if the boolean expression is true */
}
```

If the Boolean expression evaluates to **true**, then the block of code inside the 'if' statement will be executed. If the Boolean expression evaluates to **false**, then the first set of code after the end of the 'if' statement (after the closing curly brace) will be executed.

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.

Flow Diagram:



Example:

```
#include <stdio.h>

int main () {

  /* local variable definition */
```

```
   int a = 10;

 /* check the boolean condition using if statement */

         if( a < 20 ) {

     /* if condition is true then print the following */

     printf("a is less than 20\n" );

   }

    printf("value of a is : %d\n", a);

return 0;

}
```

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

Result

```
a is less than 20;

value of a is : 10;
```

1) IF....ELSE:

An **if** statement can be followed by an optional **else** statement, which executes when the Boolean expression is false.

Syntax:

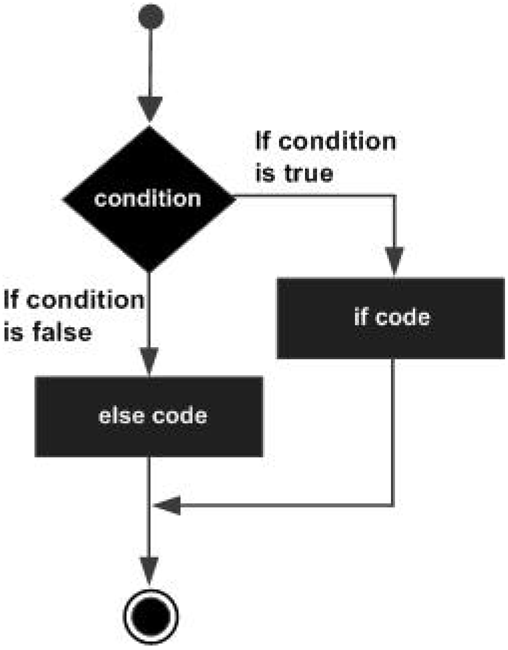
The syntax of an **if...else** statement in C programming language is −

```
if(boolean_expression) {

   /* statement(s) will execute if the boolean expression is true */

} else {

   /* statement(s) will execute if the boolean expression is false */

}
```

If the Boolean expression evaluates to **true**, then the **if block** will be executed, otherwise, the **else block** will be executed.

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.

Flow Diagram:



OPERATORS IN C:

Operators are the foundation of any programming language. Thus the functionality of C/C++ programming language is incomplete without the use of operators. We can define operators as symbols that helps us to perform specific mathematical and logical computations on operands. In other words we can say that an operator operates the operands.

For example, consider the below statement:

c = a + b;

Here, '+' is the operator known as addition operator and 'a' and 'b' are operands. The addition operator tells the compiler to add both of the operands 'a' and 'b'. C/C++ has many built-in operator types and they can be classified as:

Arithmetic Operators: These are the operators used to perform arithmetic/mathematical operations on operands. Examples: (+, -, *, /, %,++,–).

Arithmetic operator are of two types:

Unary Operators: Operators that operates or works with a single operand are unary operators.

For example: (++ , –)

Binary Operators: Operators that operates or works with two operands are binary operators.For example: (+ , – , * , /)

Relational Operators: Relational operators are used for comparison of the values of two operands. For example: checking if one operand is equal to the other operand or not, an operand is greater than the other operand or not etc. Some of the relational operators are (==, > , = , <=).

Logical Operators: Logical Operators are used to combine two or more conditions/constraints or to complement the evaluation of the original condition in consideration. The result of the operation of a logical operator is a boolean value either true or false.

Bitwise Operators: The Bitwise operators is used to perform bit-level operations on the operands. The operators are first converted to bit-level and then calculation is performed on the operands. The mathematical operations such as addition , subtraction , multiplication etc. can be performed at bit-level for faster processing.

Assignment Operators: Assignment operators are used to assign value to a variable. The left side operand of the assignment operator is a variable and right side operand of the assignment operator is a value. The value on the right side must be of the same data-type of variable on the left side otherwise the compiler will raise an error.

Different types of assignment operators are shown below:

“=”: This is the simplest assignment operator. This operator is used to assign the value on the right to the variable on the left.

For example:

- a = 10;

- b = 20;
- ch = 'y';
- "+=":This operator is combination of '+' and '=' operators. This operator first adds the current value of the variable on left to the value on right and then assigns the result to the variable on the left.
  Example:
- (a += b) can be written as (a = a + b)

If initially value stored in a is 5. Then (a += 6) = 11.

- "-=":This operator is combination of '-' and '=' operators. This operator first subtracts the current value of the variable on left from the value on right and then assigns the result to the variable on the left.
  Example:
- (a -= b) can be written as (a = a - b)

If initially value stored in a is 8. Then (a -= 6) = 2.

- "*=":This operator is combination of '*' and '=' operators. This operator first multiplies the current value of the variable on left to the value on right and then assigns the result to the variable on the left.
  Example:
- (a *= b) can be written as (a = a * b)

If initially value stored in a is 5. Then (a *= 6) = 30.

- "/=":This operator is combination of '/' and '=' operators. This operator first divides the current value of the variable on left by the value on right and then assigns the result to the variable on the left.
  Example:
- (a /= b) can be written as (a = a / b)

If initially value stored in a is 6. Then (a /= 2) = 3.

- Other Operators: Apart from the above operators there are some other operators available in C or C++ used to perform some specific task. Some of them are discussed here:

- **sizeof operator:** sizeof is a much used in the C/C++ programming language. It is a compile time unary operator which can be used to compute the size of its operand. The result of sizeof is of unsigned integral type which is usually denoted by size_t. Basically, sizeof operator is used to compute the size of the variable.
- **Comma Operator:** The comma operator (represented by the token ,) is a binary operator that evaluates its first operand and discards the result, it then evaluates the second operand and returns this value (and type). The comma operator has the lowest precedence of any C operator. Comma acts as both operator and separator
- **Conditional Operator:** Conditional operator is of the form *Expression1 ? Expression2 : Expression3* . Here, Expression1 is the condition to be evaluated. If the condition(Expression1) is *True* then we will execute and return the result of Expression2 otherwise if the condition(Expression1) is *false* then we will execute and return the result of Expression3. We may replace the use of if..else statements by conditional operators.

Operator precedence chart:

The below table describes the precedence order and associativity of operators in C / C++ . Precedence of operator decreases from top to bottom.

OPERATOR	DESCRIPTION	ASSOCIATIVITY
()	Parentheses (function call)	left-to-right
[]	Brackets (array subscript)	
.	Member selection via object name	
->	Member selection via pointer	
++/–	Postfix increment/decrement	
++/–	Prefix increment/decrement	right-to-left
+/-	Unary plus/minus	
!~	Logical negation/bitwise complement	
(type)	Cast (convert value to temporary value of type)	

*	Dereference	
&	Address (of operand)	
sizeof	Determine size in bytes on this implementation	
*,/,%	Multiplication/division/modulus	left-to-right
+/-	Addition/subtraction	left-to-right
<>	Bitwise shift left, Bitwise shift right	left-to-right
< , <=	Relational less than/less than or equal to	left-to-right
> , >=	Relational greater than/greater than or equal to	left-to-right
== , !=	Relational is equal to/is not equal to	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
|	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
||	Logical OR	left-to-right
?:	Ternary conditional	right-to-left
=	Assignment	right-to-left
+= , -=	Addition/subtraction assignment	
*= , /=	Multiplication/division assignment	
%= , &=	Modulus/bitwise AND assignment	
^= , |=	Bitwise exclusive/inclusive OR assignment	
<>=	Bitwise shift left/right assignment	
,	expression separator	left-to-right

TYPES OF LOOPS:

Loops are control structures used to repeat a given section of code a certain number of times or until a particular condition is met. Visual Basic has three main types of loops: for..next loops, do loops and while loops.

Note: 'Debug' may be a reserved word in Visual Basic, and this may cause the code samples shown here to fail for some versions of Visual Basic.

For..Next Loops

The syntax of a For..Next loop has three components: a counter, a range, and a step. A basic for..next loop appears as follows:

```
For X = 1 To 100

 Step 2

  Debug.Print X

 Next X
```

In this example,

X is the counter, "1 to 100" is the range, and "2" is the step.

The variable reference in the Next part of the statement is optional and it is common practice to leave this out. There is no ambiguity in doing this if code is correctly indented.

When a For..Next loop is initialized, the counter is set to the first number in the range; in this case, X is set to 1. The program then executes any code between the for and next statements normally. Upon reaching the next statement, the program returns to the for statement and increases the value of the counter by the step. In this instance, X will be increased to 3 on the second iteration, 5 on the third, etc.

To change the amount by which the counter variable increases on each iteration, simply change the value of the step. For example, if you use a step 3, X will increase from 1 to 4, then to 7, 10, 13, and so on. When the step is not explicitly stated, 1 is used by default. (Note that the step can be a negative value. For instance, for X = 100 to 1 step -1 would decrease the value of X from 100 to 99 to 98, etc.)

When X reaches the end of the range in the range (100 in the example above), the loop will cease to execute, and the program will continue to the code beyond the next statement.

It is possible to edit the value of the counter variable within a for..next loop, although this is generally considered bad programming practice:

```
For X = 1 To 100 Step 1

 Debug.Print X

 X = 7

Next
```

While you may on rare occasions find good reasons to edit the counter in this manner, the example above illustrates one potential pitfall:

Because X is set to 7 at the end of every iteration, this code creates an infinite loop. To avoid this and other unexpected behavior, use extreme caution when editing the counter variable!

It is not required by the compiler that you specify the name of the loop variable in the Next statement but it will be checked by the compiler if you do, so it is a small help in writing correct programs. For loop on list

Another very common situation is the need for a loop which enumerates every element of a list. The following sample code shows you how to do this:

```
Dim v As Variant

For Each v In list

 Debug.Print v

Next
```

The list is commonly a Collection or Array, but can be any other object that implements an enumerator. Note that the iterating variable has to be either a Variant, Object or class that matches the type of elements in the list.

Do Loops:

Do loops are a bit more flexible than For loops, but should generally only be used when necessary. Do loops come in the following formats:

- Do while
- Do until
- Loop while
- Loop until

While loops (both do while and loop while) will continue to execute as long as a certain conditional is true. An Until loop will loop as long as a certain condition is false, on the other hand. The only difference between putting either While or Until in the Do section or the Loop section, is that Do checks when the loop starts, and Loop checks when the loop ends. An example of a basic loop is as follows:

```
Do:

 Debug.Print "hello"

 x = x + 1

Loop Until x = 10
```

This loop will print hello several times, depending on the initial value of x. As you may have noticed, Do loops have no built in counters. However, they may be made manually as shown above. In this case, I chose x as my counter variable, and every time the loop execute, x increase itself by one. When X reaches 10, the loop will cease to execute. The advantage of Do loops is that you may exit at any time whenever any certain conditional is met. You may have it loop as long as a certain variable is false, or true, or as long as a variable remains in a certain range.

Endless loop: Do..Loop

The endless loop is a loop which never ends and the statements inside are repeated forever. Never is meant as a relative term here - if the computer is switched off then even endless loops will end very abruptly.

```
Do

 Do_Something

Loop
```

In Visual Basic you cannot label the loop but you can of course place a label just before it, inside it or just after it if you wish.

Loop with condition at the beginning:

Do While..Loop:

This loop has a condition at the beginning. The statements are repeated as long as the condition is met. If the condition is not met at the very beginning then the statements inside the loop are never executed.

```
Do While X <= 5

 X = X + 5

Loop
```

Loop with condition at the end: Do..Loop Until

This loop has a condition at the end and the statements are repeated until the condition is met. Since the check is at the end the statements are at least executed once.

```
Do

 X = 5+2

Loop Until X > 5
```

Loop with condition in the middle:Do..Exit Do..Loop

Sometimes you need to first make a calculation and exit the loop when a certain criterion is met. However when the criterion is not met there is something else to be done. Hence you need a loop where the exit condition is in the middle.

```
Do
```

```
 X = Calculate_Something

 If X > 10 then

  Exit Do

 End If

 Do_Something (X)

Loop
```

In Visual Basic you can also have more than one exit statement. You cannot exit named outer loops using Exit Do because Visual Basic does not provide named loops; you can of course use Goto instead to jump to a label that follows the outer loop.

While Loops:

While loops are similar to Do loops except that the tested condition always appears at the top of the loop. If on the first entry into the loop block the condition is false, the contents of the loop are never executed. The condition is retested before every loop iteration.

An example of a While loop is as follows:

```
price = 2

 While price < 64

        Debug.Print "Price = " & price

        price = price ^ 2

    Wend

 Debug.Print "Price = " & price & ":  Too much for the market to bear!"
```

The While loop will run until the condition tests false - or until an "Exit While" statement is encountered.

Nested Loops:

A nested loop is any type of loop inside an already existing loop. They can involve any type of loop. For this, we will use For loops. It is important to remember that the inner loop will execute its normal amount multiplied by how many times the outer loop runs. For example:

```
For i = 1 To 10

 For j = 1 To 2

  Debug.Print "hi"

 Next
```

This will print the word 'hi' twenty times. Upon the first pass of the i loop, it will run the j loop twice. Then on the second pass of the i loop, it will run the j loop another two times, and so on.

Case and Switch:

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case.

Syntax

The syntax for a switch statement in C programming language is as follows −

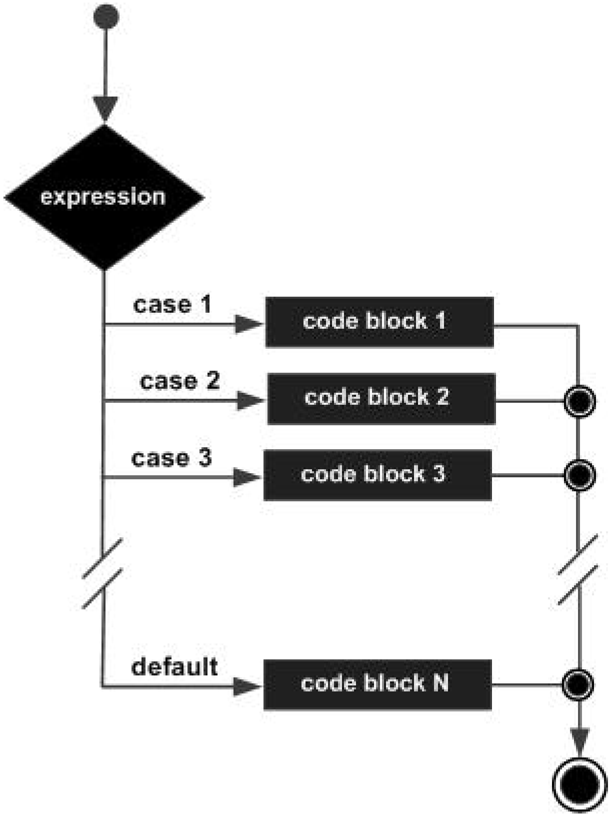
```
switch(expression) {

  case constant-expression  :

      statement(s);

      break; /* optional */


   case constant-expression  :

      statement(s);

      break; /* optional */
   /* you can have any number of case statements */

   default : /* Optional */

   statement(s);

}
```

The following rules apply to a switch statement

The expression used in a switch statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type .You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon. The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.

A switch statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

Flow Diagram



Switch statement in C:

Example

Live Demo

```
#include <stdio.h>

int main () {

 /* local variable definition */

   char grade = 'B';

 switch(grade) {

      case 'A' :

         printf("Excellent!\n" );

         break;
```

```
    case 'B' :
    case 'C' :
      printf("Well done\n" );
      break;
    case 'D' :
      printf("You passed\n" );
      break;
    case 'F' :
      printf("Better try again\n" );
      break;
    default :
printf("Invalid grade\n" );
  }
printf("Your grade is  %c\n", grade );
return 0;
}
```

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

Result :

```
Well done
Your grade is B
```

Arrays and Functions:

- Arrays
- C Functions – Call by values and Call by reference

Arrays:

Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

Declaring Arrays:

To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows –

```
type arrayName [ arraySize ];
```

This is called a *single-dimensional* array. The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type. For example, to declare a 10-element array called **balance** of type double, use this statement −

```
double balance[10];
```

Here *balance* is a variable array which is sufficient to hold up to 10 double numbers.

Initializing Arrays:

You can initialize an array in C either one by one or using a single statement as follows −

```
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

The number of values between braces { } cannot be larger than the number of elements that we declare for the array between square brackets [ ].

If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write −

```
double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

You will create exactly the same array as you did in the previous example. Following is an example to assign a single element of the array −

```
balance[4] = 50.0;
```

The above statement assigns the 5th element in the array with a value of 50.0. All arrays have 0 as the index of their first element which is also called the base index and the last index of an array will be total size of the array minus 1. Shown below is the pictorial representation of the array we discussed above .

Accessing Array Elements:

An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example −

double salary = balance[9]; The above statement will take the 10th element from the array and assign the value to salary variable. The following example Shows how to use all the three above mentioned concepts viz. declaration, assignment, and accessing arrays .

Arrays in Detail

Arrays are important to C and should need a lot more attention. The following important concepts related to array should be clear to a C programmer −

Sr.No.	Concept & Description
1	**Multi-dimensional arrays** C supports multidimensional arrays. The simplest form of the multidimensional array is the two-dimensional array.
2	**Passing arrays to functions** You can pass to the function a pointer to an array by specifying the array's name without an index.
3	**Return array from a function** C allows a function sto return an array.
4	**Pointer to an array** You can generate a pointer to the first element of an array by simply specifying the array name, without any index.

Call by Values and Call by Reference:

There are two ways to pass arguments/parameters to function calls -- *call by value* and *call by reference*. The major difference between call by value and call by reference is that in call by value a copy of actual arguments is passed to respective formal arguments. While, in call by reference the location (address) of actual arguments is passed to formal arguments, hence any change made to formal arguments will also reflect in actual arguments.

In C, all function arguments are passed "by value" because C does not support references like C++ and Java do. In C, the calling and called functions do not share any memory -- they have their own copy and the called function cannot directly alter a variable in the calling function; it can only alter its private, temporary copy.

The *call by value* scheme is an asset, however, not a liability. It usually leads to more compact programs with fewer extraneous variables, because parameters can be treated as conveniently initialized local variables in the called routine. Yet, there are some cases where we need *call by reference*:

Difference between *call by value* and *call by reference*

call by value	call by reference
In *call by value*, a copy of actual arguments is passed to formal arguments of the called function and any change made to the formal arguments in the called function have no effect on the values of actual arguments in the calling function.	In *call by reference*, the location (address) of actual arguments is passed to formal arguments of the called function. This means by accessing the addresses of actual arguments we can alter them within from the called function.
In call by value, actual arguments will remain safe, they cannot be modified accidentally.	In *call by reference*, alteration to actual arguments is possible within from called function; therefore the code must handle arguments carefully else you get unexpected results.

Data Base Management System:-

A database is an organized collection of data.A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Computer scientists may classify database-management systems according to the database models that they support; the most popular database systems since the 1980s have all supported the relational model - generally associated with the SQL language.[disputed – discuss] Sometimes a DBMS is loosely referred to as a "database".

A **relational database** is a digital database based on the relational model of data, as proposed by E. F. Codd in 1970. A software system used to maintain relational databases is a relational database management system (RDBMS). Virtually all relational database systems use SQL (Structured Query Language) for querying and maintaining the database.

Data Base Tables:-

A table is an arrangement of data in rows and columns, or possibly in a more complex structure. Tables are widely used in communication, research, and data analysis. Tables appear in print media, handwritten notes, computer software, architectural ornamentation, traffic signs, and many other places. The precise conventions and terminology for describing tables vary depending on the context. Further, tables differ significantly in variety, structure, flexibility, notation, representation and use. In books and technical articles, tables are typically presented apart from the main text in numbered and captioned floating blocks.

A **database** is an organized collection of data.[1] A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Computer scientists may classify database-management systems according to the database models that they support; the most popular database systems since the 1980s have all supported the relational model - generally associated with the SQL language.[*disputed – discuss*] Sometimes a DBMS is loosely referred to as a "database".

In a relational database, a **column** is a set of data values of a particular simple type, one value for each row of the database. A column may contain text values, numbers, or even pointers to files in the operating system. Some relational database systems allow columns to contain more complex data types; whole documents, images or even video clips are examples. A column can also be called an **attribute**.

Each row would provide a data value for each column and would then be understood as a single structured data value. For example, a database that represents company contact information might have the following columns: ID, Company Name, Address Line 1, Address Line 2, City, and Postal Code. More formally, each row can be interpreted as a relvar, composed of a set of tuples, with each tuple consisting of the relevant column and its value, for example, the tuple ('Address 1', '12345 West Example Street').

In the context of a relational database, a **row**—also called a record or tuple—represents a single, implicitly structured data item in a table. In simple terms, a database table can be thought of as consisting of *rows* and columns or fields.[1] Each row in a table represents a set of related data, and every row in the table has the same structure.

For example, in a table that represents companies, each row would represent a single company. Columns might represent things like company name, company street address, whether the company is publicly held, its VAT number, etc.. In a table that represents *the association* of employees with departments, each row would associate one employee with one department.

In a less formal usage, e.g. for a database which is not formally relational, a record is equivalent to a row as described above, but is not usually referred to as a row.

The implicit structure of a row, and the meaning of the data values in a row, requires that the row be understood as providing a succession of data values, one in each column of the table. The row is then interpreted as a relvar composed of a set of tuples, with each tuple consisting of the two items: the name of the relevant column and the value this row provides for that column.

Each column expects a data value of a particular type. For example, one column might require a unique identifier, another might require text representing a person's name, another might require an integer representing hourly pay in cents.

A **flat file database** is a database stored as an ordinary unstructured file called a "flat file". To access the structure of the data and manipulate it on a computer system, the file must be read in its entirety into the computer's memory. Upon completion of the database operations, the file is again written out in its entirety to the host's file system. In this stored mode the database is said to be "flat", meaning that it has no structure for indexing and there are usually no structural relationships between the records. A flat file can be a plain text file or a binary file.

The term has generally implied a small, simple database. As computer memory has become cheaper, more sophisticated databases can now be entirely held in memory for faster access. These newer databases would not generally be referred to as flat-file databases.

Query Basics and SQL:

SQL, Structured Query Language, is a programming language designed to manage data stored in relational databases. SQL operates through simple, declarative statements. This keeps data accurate and secure, and it helps maintain the integrity of databases, regardless of size. Here's an appendix of commonly used commands.

Commands

ALTER TABLE

```
ALTER TABLE table_name
ADD column_name datatype;
```

ALTER TABLE lets you add columns to a table in a database.

AND

```
SELECT column_name(s)
FROM table_name
WHERE column_1 = value_1
```

```
  AND column_2 = value_2;
```

`AND` is an operator that combines two conditions. Both conditions must be true for the row to be included in the result set.

AS

```
SELECT column_name AS 'Alias'
FROM table_name;
```

AS is a keyword in SQL that allows you to rename a column or table using an *alias*.

AVG()

```
SELECT AVG(column_name)
FROM table_name;
```

`AVG()` is an aggregate function that returns the average value for a numeric column.

BETWEEN

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value_1 AND value_2;
```

The `BETWEEN` operator is used to filter the result set within a certain range. The values can be numbers, text or dates.

CASE

```
SELECT column_name,
  CASE
    WHEN condition THEN 'Result_1'
    WHEN condition THEN 'Result_2'
    ELSE 'Result_3'
  END
FROM table_name;
```

`CASE` statements are used to create different outputs (usually in the `SELECT` statement). It is SQL's way of handling if-then logic.

COUNT()

```
SELECT COUNT(column_name)
FROM table_name;
```

`COUNT()` is a function that takes the name of a column as an argument and counts the number of rows where the column is not `NULL`.

CREATE TABLE

```
CREATE TABLE table_name (
  column_1 datatype,
  column_2 datatype,
  column_3 datatype
);
```

`CREATE TABLE` creates a new table in the database. It allows you to specify the name of the table and the name of each column in the table.

DELETE

```
DELETE FROM table_name
WHERE some_column = some_value;
```

DELETE statements are used to remove rows from a table.

GROUP BY

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name;
```

`GROUP BY` is a clause in SQL that is only used with aggregate functions. It is used in collaboration with the `SELECT` statement to arrange identical data into groups.

HAVING

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name
HAVING COUNT(*) > value;
```

`HAVING` was added to SQL because the `WHERE` keyword could not be used with aggregate functions.

INNER JOIN

```
SELECT column_name(s)
FROM table_1
JOIN table_2
  ON table_1.column_name = table_2.column_name;
```

An inner join will combine rows from different tables if the *join condition* is true.

INSERT

```
INSERT INTO table_name (column_1, column_2, column_3)
VALUES (value_1, 'value_2', value_3);
```

`INSERT` statements are used to add a new row to a table.

IS NULL / IS NOT NULL

```
SELECT column_name(s)
FROM table_name
WHERE column_name IS NULL;
```

`IS NULL` and `IS NOT NULL` are operators used with the `WHERE` clause to test for empty values.

LIKE

```
SELECT column_name(s)
FROM table_name
WHERE column_name LIKE pattern;
```

`LIKE` is a special operator used with the `WHERE` clause to search for a specific pattern in a column.

LIMIT

```
SELECT column_name(s)
FROM table_name
LIMIT number;
```

`LIMIT` is a clause that lets you specify the maximum number of rows the result set will have.

MAX()

```
SELECT MAX(column_name)
FROM table_name;
```

`MAX()` is a function that takes the name of a column as an argument and returns the largest value in that column.

MIN()

```
SELECT MIN(column_name)
FROM table_name;
```

`MIN()` is a function that takes the name of a column as an argument and returns the smallest value in that column.

OR

```
SELECT column_name
FROM table_name
WHERE column_name = value_1
   OR column_name = value_2;
```

`OR` is an operator that filters the result set to only include rows where either condition is true.

ORDER BY

```
SELECT column_name
FROM table_name
ORDER BY column_name ASC | DESC;
```

`ORDER BY` is a clause that indicates you want to sort the result set by a particular column either alphabetically or numerically.

OUTER JOIN

```
SELECT column_name(s)
```

```
FROM table_1
LEFT JOIN table_2
  ON table_1.column_name = table_2.column_name;
```

An outer join will combine rows from different tables even if the join condition is not met. Every row in the *left* table is returned in the result set, and if the join condition is not met, then `NULL` values are used to fill in the columns from the *right* table.

ROUND()

```
SELECT ROUND(column_name, integer)
FROM table_name;
```

`ROUND()` is a function that takes a column name and an integer as an argument. It rounds the values in the column to the number of decimal places specified by the integer.

SELECT

```
SELECT column_name
FROM table_name;
```

SELECT statements are used to fetch data from a database. Every query will begin with SELECT.

SELECT DISTINCT

```
SELECT DISTINCT column_name
FROM table_name;
```

`SELECT DISTINCT` specifies that the statement is going to be a query that returns unique values in the specified column(s).

SUM

```
SELECT SUM(column_name)
FROM table_name;
```

`SUM()` is a function that takes the name of a column as an argument and returns the sum of all the values in that column.

UPDATE

```
UPDATE table_name
SET some_column = some_value
WHERE some_column = some_value;
```

`UPDATE` statements allow you to edit rows in a table.

WHERE

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator value;
```

`WHERE` is a clause that indicates you want to filter the result set to include only rows where the following *condition* is true.

Visual Basics (VB)

Introduction to Visual Basics Controls:-

Visual Basic is a third-generation event-driven programming language and integrated development environment (IDE) from Microsoft for its Component Object Model (COM) programming model first released in 1991 and declared legacy during 2008. Microsoft intended Visual Basic to be relatively easy to learn and use.Visual Basic was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects.

A programmer can create an application using the components provided by the Visual Basic program itself. Over time the community of programmers developed third-party components .Programs written in Visual Basic can also use the Windows API, which requires external function declarations.

The final release was version 6 in 1998 (now known simply as Visual Basic). On April 8, 2008, Microsoft stopped supporting Visual Basic 6.0 IDE. The Microsoft Visual Basic team still maintains compatibility for Visual Basic 6.0 applications on Windows Vista, Windows Server 2008 including R2, Windows 7, Windows 8, Windows 8.1, Windows Server 2012 and Windows 10 through its "It Just Works" program .In 2014, some software developers still preferred Visual Basic 6.0 over its successor, Visual Basic .NET .In 2014 some developers lobbied for a new version of Visual Basic 6.0. In 2016, Visual Basic 6.0 won the technical impact award at The 19th Annual D.I.C.E. Awards .A dialect of Visual Basic, Visual Basic for Applications (VBA), is used as a macro or scripting language within several Microsoft applications, including Microsoft Office.

Third-generation programming language

3GLs are much more machine independent and more programmer-friendly. This includes features like improved support for aggregate data types, and expressing concepts in a way that favors the programmer, not the computer. A third generation language improves over a second generation language by having the computer take care of non-essential details. 3GLs feature more abstraction than previous generations of languages, and thus can be considered higher level languages than their first and second generation counterparts. First introduced in the late 1950s, Fortran, ALGOL, and COBOL are examples of early 3GLs.

Most popular general-purpose languages today, such as C, C++, C#, Java, BASIC and Pascal, are also third-generation languages, although each of these languages can be further subdivided into other categories based on other contemporary traits. Most 3GLs support structured programming. Many support object-oriented programming. Traits like these are more often used to describe a language rather than just being a 3GL.

A programming language such as C, FORTRAN, or Pascal enables a programmer to write programs that are more or less independent from a particular type of computer. Such languages are considered high-level because they are closer to human languages and further from machine languages, and hence require compilation or interpretation.

In contrast, machine languages are considered as low-level because they are designed for and executed by physical hardware without further translation required.

The main advantage of high-level languages over low-level languages is that they are easier to read, write, and maintain. Ultimately, programs written in a high-level language must be translated into machine language by a compiler or directly into behaviour by an interpreter.

These programs could run on different machines so they were machine-independent. As new, more abstract languages have been developed, however, the concept of high and low level languages have become rather relative. Many of the early "high level" languages are now considered relatively low level in comparison to languages such as Python, Ruby, and Common Lisp, which have some features of fourth-generation programming languages.

VB Interface:-

Interfaces define the properties, methods, and events that classes can implement. Interfaces allow you to define features as small groups of closely related properties, methods, and events; this reduces compatibility problems because you can develop enhanced implementations for your interfaces without jeopardizing existing code. You can add new features at any time by developing additional interfaces and implementations.

There are several other reasons why you might want to use interfaces instead of class inheritance:

- Interfaces are better suited to situations in which your applications require many possibly unrelated object types to provide certain functionality.
- Interfaces are more flexible than base classes because you can define a single implementation that can implement multiple interfaces.
- Interfaces are better in situations in which you do not have to inherit implementation from a base class.
- Interfaces are useful when you cannot use class inheritance. For example, structures cannot inherit from classes, but they can implement interfaces.

Declaring Interfaces

Interface definitions are enclosed within the `Interface` and `End Interface` statements. Following the `Interface` statement, you can add an optional `Inherits` statement that lists one or more inherited interfaces. The `Inherits` statements must precede all other statements in the declaration except comments. The remaining statements in the interface definition should be `Event`, `Sub`, `Function`, `Property`, `Interface`, `Class`, `Structure`, and `Enum` statements. Interfaces cannot contain any implementation code or statements associated with implementation code, such as `End Sub` or `End Property`.

In a namespace, interface statements are `Friend` by default, but they can also be explicitly declared as `Public` or `Friend`. Interfaces defined within classes, modules, interfaces, and structures are `Public` by default, but they can also be explicitly declared as `Public`, `Friend`, `Protected`, or `Private`.

For example, the following code defines an interface with one function, one property, and one event.

VB

```
Interface IAsset
   Event ComittedChange(ByVal Success As Boolean)
   Property Division() As String
   Function GetID() As Integer
End Interface.
```

Implements Statement:

If a class or structure implements one or more interfaces, it must include the Implements statement immediately after the Class or Structure statement. The Implements statement requires a comma-separated list of interfaces to be implemented by a class. The class or structure must implement all interface members using the Implements keyword.

Implements Keyword:

The Implements keyword requires a comma-separated list of interface members to be implemented. Generally, only a single interface member is specified, but you can specify multiple members. The specification of an interface member consists of the interface name, which must be specified in an implements statement within the class; a period; and the name of the member function, property, or event to be implemented. The name of a member that implements an interface member can use any legal identifier, and it is not limited to the Interface Name_ Method Name convention used in earlier versions of Visual Basic.

For example, the following code shows how to declare a subroutine named Sub1 that

implements a method of an interface:

VB

```
Class Class1
   Implements interfaceclass.interface2

   Sub Sub1(ByVal i As Integer) Implements interfaceclass.interface2.Sub1
   End Sub
End Class
```

The parameter types and return types of the implementing member must match the interface property or member declaration in the interface. The most common way to implement an element of an interface is with a member that has the same name as the interface, as shown in the previous example.

To declare the implementation of an interface method, you can use any attributes that are legal on instance method declarations, including Overloads, Overrides, Overridable, Public, Private, Protected, Friend, Protected Friend, MustOverride, Default, and Static. The Shared attribute is not legal since it defines a class rather than an instance method.

Using Implements, you can also write a single method that implements multiple methods

defined in an interface, as in the following example:

VB

```
Class Class2
```

```
    Implements I1, I2

    Protected Sub M1() Implements I1.M1, I1.M2, I2.M3, I2.M4
    End Sub
End Class
```

You can use a private member to implement an interface member. When a private member implements a member of an interface, that member becomes available by way of the interface even though it is not available directly on object variables for the class.

Interface Implementation Examples

Classes that implement an interface must implement all its properties, methods, and events.

The following example defines two interfaces. The second interface, Interface2, inherits Interface1 and defines an additional property and method.

VB

```
Interface Interface1
    Sub sub1(ByVal i As Integer)
End Interface

' Demonstrates interface inheritance.

Interface Interface2
    Inherits Interface1
    Sub M1(ByVal y As Integer)
    ReadOnly Property Num() As Integer
End Interface
```

The next example implements Interface1, the interface defined in the previous example:

VB

```
Public Class ImplementationClass1
    Implements Interface1
    Sub Sub1(ByVal i As Integer) Implements Interface1.sub1
        ' Insert code here to implement this method.
    End Sub
End Class
```

The final example implements Interface2, including a method inherited from Interface1:

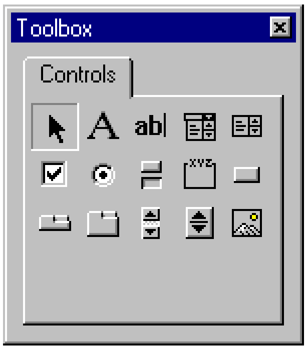
VB

```
Public Class ImplementationClass2
    Implements Interface2
    Dim INum As Integer = 0
    Sub sub1(ByVal i As Integer) Implements Interface2.sub1
        ' Insert code here that implements this method.
    End Sub
    Sub M1(ByVal x As Integer) Implements Interface2.M1
        ' Insert code here to implement this method.
    End Sub
    ReadOnly Property Num() As Integer Implements Interface2.Num
        Get
            Num = INum
        End Get
    End Property
End Class
```

You can implement a readonly property with a readwrite property (that is, you do not have to declare it readonly in the implementing class). Implementing an interface promises to implement at least the members that the interface declares, but you can offer more functionality, such as allowing your property to be writable.

Tool Box:

Displays the standard Visual Basic controls plus any ActiveX controls and insertable objects you have added to your project.



You can customize the Toolbox by adding pages to it or by adding controls using the Additional Controls command from the Tools menu.

When you add a page, a Select Objects tool is always available on the page.
Standard Toolbox Controls

Select Objects
Select Objects is the only item in the Toolbox that doesn't draw a control. When you select it, you can only resize or move a control that has already been drawn on a form.

Label
Allows you to have text that you do not want the user to change, such as a caption under a graphic.

Text Box
Holds text that the user can either enter or change.

Combo Box
Allows you to draw a combination list box and text box. The user can either choose an item from the list or enter a value in the text box.

List Box
Use to display a list of items from which the user can choose. The list can be scrolled if it has more items than can be displayed at one time.

Check Box
Creates a box that the user can easily choose to indicate if something is true or false, or to display multiple choices when the user can choose more than one.

Option Button
Allows you to display multiple choices from which the user can choose only one.

Toggle Button
Creates a button that toggles on and off.

Frame
Allows you to create a graphical or functional grouping for controls. To group controls, draw the frame first, and then draw controls inside the frame.

Command Button
Creates a button the user can choose to carry out a command.

Tab Strip
Allows you to define multiple pages for the same area of a window or dialog box in your application.

Multi Page
Presents multiple screens of information as a single set.

Scroll Bar
Provides a graphical tool for quickly navigating through a long list of items or a large amount of information, for indicating the current position on a scale, or as an input device or indicator of speed or quantity.

Spin Button
A spinner control you can use with another control to increment and decrement numbers. You can also use it to scroll back and forth through a range of values or a list of items.

Image
Displays a graphical image from a bitmap, icon, or metafile on your form. Images displayed in an Image control can only be decorative and use fewer resources than a Picture Box.

Object Browser Control:-

The **Object Browser** allows you to browse through all available objects in your project and see their properties, methods and events. In addition, you can see the procedures and constants that are available from object libraries in your project. You can easily display online Help as you browse. You can use the **Object Browser** to find and use objects you create, as well as objects from other applications.

You can get help for the **Object Browser** by searching for **Object Browser** in Help.

To navigate the Object Browser

1. Activate a module.
2. From the **View** menu, choose **Object Browser** (F2), or use the toolbar shortcut: .
3. Select the name of the project or library you want to view in the **Project/Library** list.
4. Use the **Class** list to select the class; use the **Member** list to select specific members of your class or project.
5. View information about the class or member you selected in the **Details** section at the bottom of the window.
6. Use the **Help** button to display the Help topic for the class or member you selected.

VBE Glossary:

1) access key

A key pressed while holding down the ALT key that allows the user to open a menu, carry out a command, select an object, or move to an object. For example, ALT+F opens the **File** menu.

2) ActiveX control

An object that you place on a form to enable or enhance a user's interaction with an application. ActiveX Controls have events and can be incorporated into other controls. These controls have an . ocx file name extension.

3) ActiveX object

An object that is exposed to other applications or programming tools through Automation interfaces.

4) add-in

A customized tool that adds capabilities to the Visual Basic development environment.

5) ANSI Character Set

American National Standards Institute (ANSI) 8-bit character set used to represent up to 256 characters (0-255) using your keyboard. The first 128 characters (0-127) correspond to the letters and symbols on a standard U.S. keyboard. The second 128 characters (128-255) represent special characters, such as letters in international alphabets, accents, currency symbols, and fractions.

6) **application**

A collection of code and visual elements that work together as a single program. Developers can build and run applications within the development environment, while users usually run applications as executable files outside the development environment.

7) **argument**

A constant, variable, or expression passed to a procedure.

8) **array**

A set of sequentially indexed elements having the same intrinsic data type. Each element of an array has a unique identifying index number. Changes made to one element of an array don't affect the other elements.

9) **ASCII Character Set**

American Standard Code for Information Interchange (ASCII) 7-bit character set used to represent letters and symbols found on a standard U.S. keyboard. The ASCII character set is the same as the first 128 characters (0-127) in the ANSI character set.

10) **automatic formatting**

A feature that automatically formats code as you enter it by capitalizing the first letter for keywords, standardizing spacing, adding punctuation, and setting the foreground and background colors.

11) **Automation object**

An object that is exposed to other applications or programming tools through Automation interfaces.

12) **base class**

Original class from which other classes can be derived by inheritance.

13) **bitmap**

An image represented by pixels and stored as a collection of bits in which each bit corresponds to one pixel. On color systems, more than one bit corresponds to each pixel. A bitmap usually has a .bmp file name extension.

14) bitwise comparison

A bit-by-bit comparison between identically positioned bits in two numeric expressions.

15) Boolean expression

An expression that evaluates to either **True** or **False**.

16) Boolean data type

A data type with only two possible values, **True** (-1) or **False** (0). **Boolean** variables are stored as 16-bit (2-byte) numbers.

17) bound control

A data-aware control that can provide access to a specific field or fields in a database through a **Data** control. A data-aware control is typically bound to a **Data** control through its **DataSource** and **DataField** properties. When a **Data** control moves from one record to the next, all bound controls connected to the **Data** control change to display data from fields in the current record. When users change data in a bound control and then move to a different record, the changes are automatically saved in the database.

18) break mode

Temporary suspension of program execution in the development environment. In break mode, you can examine, debug, reset, step through, or continue program execution. You enter break mode when you:

- Encounter a breakpoint during program execution.
- Press CTRL+BREAK during program execution.
- Encounter a **Stop** statement or untrapped run-time error during program execution.
- Add a **Break When True** watch expression. Execution stops when the value of the watch changes and evaluates to **True**.
- Add a **Break When Changed** watch expression. Execution stops when the value of the watch changes.

19) breakpoint

A selected program line at which execution automatically stops. Breakpoints are not saved with your code.

20) by reference

A way of passing the address of an argument to a procedure instead of passing the value. This allows the procedure to access the actual variable. As a result, the variable's actual value can be changed by the procedure to which it is passed. Unless otherwise specified, arguments are passed by reference.

21) Byte data type

A data type used to hold positive integer numbers ranging from 0-255. Byte variables are stored as single, unsigned 8-bit (1-byte) numbers.

22) by value

A way of passing the value of an argument to a procedure instead of passing the address. This allows the procedure to access a copy of the variable. As a result, the variable's actual value can't be changed by the procedure to which it is passed.

23) character code

A number that represents a particular character in a set, such as the ANSI character set.

24) class

The formal definition of an object. The class acts as the template from which an instance of an object is created at run time. The class defines the properties of the object and the methods used to control the object's behavior.

25) class module

A module that contains the definition of a class, including its property and method definitions.

26) code module

A module containing public code that can be shared among all modules in a project. A code module is referred to as a standard module in later versions of Visual Basic.

27) code pane

A pane contained in a code window that is used for entering and editing code. A code window can contain one or more code panes.

28) collection

An object that contains a set of related objects. An object's position in the collection can change whenever a change occurs in the collection; therefore, the position of any specific object in the collection can vary.

29) command line

The path, file name, and argument information provided by the user to run a program.

30) comment

Text added to code that explains how the code works. In Visual Basic, a comment line can start with either an apostrophe (**'**) or with the **Rem** keyword followed by a space.

31) comparison operator

A character or symbol indicating a relationship between two or more values or expressions. These operators include less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), not equal (<>), and equal (=). Additional comparison operators include **Is** and **Like**. Note that **Is** and **Like** can't be used as comparison operators in a **SelectCase** statement.

32) compiler directive

A command used to alter the action of the compiler.

33) compile time

The period during which source code is translated to executable code.

34) conditional compiler constant

A Visual Basic identifier that is defined using the **#Const** compiler directive or defined in the host application and used by other compiler directives to determine when or if certain blocks of Visual Basic code are compiled.

35) constant

A named item that retains a constant value throughout the execution of a program. A constant can be a string or numeric literal, another constant, or any combination that includes arithmetic or logical operators except **Is** and exponentiation. Each host application can define its own set of constants. Additional constants can be defined by the user with the **Const** statement. You can use constants anywhere in your code in place of actual values.

36) container

An object that can contain other objects.

37) control

An object you can place on a form that has its own set of recognized properties, methods, and events. You use controls to receive user input, display output, and trigger event procedures. You can manipulate most controls using methods. Some controls are interactive (responsive to user actions), while others are static (accessible only through code).

38) control array

A group of controls that share a common name, type, and event procedures. Each control in an array has a unique index number that can be used to determine which control recognizes an event.

39) Currency data type

A data type with a range of -922,337,203,685,477.5808 to 922,337,203,685,477.5807. Use this data type for calculations involving money and for fixed-point calculations where accuracy is particularly important. The at sign (@) type-declaration character represents **Currency** in Visual Basic.

40) data type

The characteristic of a variable that determines what kind of data it can hold. Data types include **Byte**, **Boolean**, **Integer**, **Long**, **Currency**, **Decimal**, **Single**, **Double**, **Date**, **String**, **Object**, **Variant** (default), and user-defined types, as well as specific types of objects.

41) Date data type

A data type used to store dates and times as a real number. Date variables are stored as 64-bit (8-byte) numbers. The value to the left of the decimal represents a date, and the value to the right of the decimal represents a time.

42) date expression

Any expression that can be interpreted as a date, including date literals, numbers that look like dates, strings that look like dates, and dates returned from functions. A date expression is limited to numbers or strings, in any combination, that can represent a date from January 1, 100 - December 31, 9999.

Dates are stored as part of a real number. Values to the left of the decimal represent the date; values to the right of the decimal represent the time. Negative numbers represent dates prior to December 30, 1899.

43) date literal

Any sequence of characters with a valid format that is surrounded by number signs (#). Valid formats include the date format specified by the locale settings for your code or the universal date format.

For example, `#12/31/92#` is the date literal that represents December 31, 1992, where English-U.S. is the locale setting for your application. Use date literals to maximize portability across national languages.

44) date separators

Characters used to separate the day, month, and year when date values are formatted. The characters are determined by system settings or by the **Format** function.

45) DBCS

A character set that uses 1 or 2 bytes to represent a character, allowing more than 256 characters to be represented.

46) dynamic data exchange (DDE)

An established protocol for exchanging data through active links between applications that run under Microsoft Windows.

47) Decimal data type

A data type that contains decimal numbers scaled by a power of 10. For zero-scaled numbers, that is, numbers with no decimal places, the range is +/-79,228,162,514,264,337,593,543,950,335. For numbers with 28 decimal places the range is +/-7.9228162514264337593543950335. The smallest non-zero number that can be represented as a **Decimal** is 0.0000000000000000000000000001.

Note that at this time the **Decimal** data type can only be used within a **Variant**. You cannot declare a variable to be of type **Decimal**. You can, however, create a **Variant** whose subtype is **Decimal** using the **CDec** function.

48) declaration

Non executable code that names a constant, variable, or procedure, and specifies its characteristics, such as data type. For DLL procedures, declarations specify names, libraries, and arguments.

49) designer

Provides a visual design window in the Visual Basic development environment. You can use this window to design new classes visually. Visual Basic has built-in designers for forms. The Professional and Enterprise editions of Visual Basic include designers for ActiveX controls and ActiveX documents.

50) design time

The time during which you build an application in the development environment by adding controls, setting control or form properties, and so on. In contrast, during run time, you interact with the application like a user.

51) development environment

The part of the application where you write code, create controls, set control and form properties, and so on. This contrasts with running the application.

52) dynamic-link library (DLL)

A library of routines loaded and linked into applications at run time. DLLs are created with other programming languages such as C, MASM, or FORTRAN.

53) docked window

A window that is attached to the frame of the main window.

54) document

Any self-contained work created with an application and given a unique file name.

55) Double data type

A data type that holds double-precision floating-point numbers as 64-bit numbers in the range -1.79769313486231E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values. The number sign (#) type-declaration character represents the **Double** in Visual Basic.

56) Empty

Indicates that no beginning value has been assigned to a **Variant** variable. An **Empty** variable is represented as 0 in a numeric context or a zero-length string ("") in a string context.

57) error number

A whole number in the range 0 - 65,535 that corresponds to the **Number** property setting of the **Err** object. When combined with the **Description** property setting of the **Err** object, this number represents a particular error message.

58) event source object

An object that is the source of events that occur in response to an action. An event source object is returned by a property. For example, the **Command Bar Events** property returns the **Command Bar Events** object.

59) executable file

A Windows-based application that can run outside the development environment. An executable file has an .exe file name extension.

60) expression

A combination of keywords, operators, variables, and constants that yields a string, number, or object. An expression can be used to perform a calculation, manipulate characters, or test data.

61) file number

Number used in the **Open** statement to open a file. Use file numbers in the range 1-255, inclusive, for files not accessible to other applications. Use file numbers in the range 256-511 for files accessible from other applications.

62) focus

The ability to receive mouse clicks or keyboard input at any one time. In the Microsoft Windows environment, only one window, form, or control can have this ability at a time. The object that "has the focus" is normally indicated by a highlighted caption or title bar. The focus can be set by the user or by the application.

63) form

A window or dialog box. Forms are containers for controls. A multiple-document interface (MDI) form can also act as a container for child forms and some controls.

64) form module

A file in a Visual Basic project with an .frm file name extension that can contain graphical descriptions of a form; its controls and their property settings; form-level declarations of constants, variables, and external procedures; and event and general procedures.

65) Function procedure

A procedure that performs a specific task within a program and returns a value. A **Function** procedure begins with a **Function** statement and ends with an **End Function** statement.

66) general procedure

A procedure that must be explicitly called by another procedure. In contrast, an event procedure is invoked automatically in response to a user or system action.

67) graphics method

A method that operates on an object such as a **Form**, **PictureBox**, or **Printer**, and performs run-time drawing operations such as animation or simulation. The graphics methods are **Circle**, **Cls**, **Line**, **PaintPicture**, **Point**, **Print**, and **PSet**.

68) host application

Any application that supports the use of Visual Basic for Applications, for example, Microsoft Excel, Microsoft Project, and so on.

69) icon

A graphical representation of an object or concept; commonly used to represent minimized applications in Microsoft Windows. An icon is a bitmap with a maximum size of 32 x 32 pixels. Icons have an .ico file name extension.

70) identifier

An element of an expression that refers to a constant or variable.

71) in process

Running in the same address space as an application.

72) insertable object

An application object that is a type of custom control, such as a Microsoft Excel worksheet.

73) Integer data type

A data type that holds integer variables stored as 2-byte whole numbers in the range -32,768 to 32,767. The **Integer** data type is also used to represent enumerated values. The percent sign (**%**) type-declaration character represents an **Integer** in Visual Basic.

74) intrinsic constants

A constant provided by an application. Visual Basic constants are listed in the object library and can be viewed with the **Object Browser**. Because you can't disable intrinsic constants, you can't create a user-defined constant with the same name.

75) keyword

A word or symbol recognized as part of the Visual Basic programming language; for example, a statement, function name, or operator.

76) line-continuation character

The combination of a space followed by an underscore (_) used in the development environment to extend a single logical line of code to two or more physical lines. However, you can't use a line-continuation character to continue a line of code within a string expression.

77) line label

Used to identify a single line of code. A line label can be any combination of characters that starts with a letter and ends with a colon (**:**). Line labels are not case sensitive and must begin in the first column.

78) line number

Used to identify a single line of code. A line number can be any combination of digits that is unique within the module where it is used. Line numbers must begin in the first column.

79) linked window

A window that is joined to another window other than the main window.

80) linked window frame

A window frame containing multiple windows that have been linked together.

81) locale

The set of information that corresponds to a given language and country/region. The code locale setting affects the language of terms such as keywords and defines locale-specific settings such as the decimal and list separators, date formats, and character sorting order.

The system locale setting affects the way locale-aware functionality behaves, for example, when you display numbers or convert strings to dates. You set the system locale using the **Control Panel** utilities provided by the operating system.

Although the code locale and system locale are generally set to the same setting, they may differ in some situations. For example, in Visual Basic, Standard Edition and Visual Basic, Professional Edition, the code is not translated from English-U.S. The system locale can be set to the user's language and country/region, but the code locale is always set to English-U.S. and can't be changed. In this case, the English-U.S. separators, format placeholders, and sorting order are used.

82) logic error

A programming error that can cause code to produce incorrect results or stop execution. For example, a logic error can be caused by incorrect variable names, incorrect variable types, endless loops, flaws in comparisons, or array problems.

83) Long data type

A 4-byte integer ranging in value from -2,147,483,648 to 2,147,483,647. The ampersand (**&;**) type-declaration character represents a **Long** in Visual Basic.

84) margin indicator

An icon displayed in the **Margin Indicator** bar in the **Code** window. Margin indicators provide visual cues during code editing.

85) MDI child

A form contained within an MDI form in a multiple-document interface (MDI) application. To create a child form, set the **MDIChild** property of the MDI form to **True**.

86) MDI form

A window that makes up the background of a multiple-document interface (MDI) application. The MDI form is the container for any MDI child forms in the application.

87) member

An element of a collection, object, or user-defined type.

88) metafile

A file that stores an image as graphical objects such as lines, circles, and polygons rather than as pixels. There are two types of metafiles, standard and enhanced. Standard metafiles usually have a .wmf file name extension. Enhanced metafiles usually have a .emf file name extension. Metafiles preserve an image more accurately than pixels when the image is resized.

89) method

A procedure that acts on an object.

90) module

A set of declarations followed by procedures.

91) module level

Describes code in the Declarations section of a module. Any code outside a procedure is referred to as module-level code. Declarations must be listed first, followed by procedures.

92) module variable

A variable declared outside **Function**, **Sub**, or **Property** procedure code. Module variables must be declared outside any procedures in the module. They exist while the module is loaded and are visible in all procedures in the module.

93) named argument

An argument that has a name that is predefined in the object library. Instead of providing a value for each argument in a specified order expected by the syntax, you can use named arguments to assign values in any order. For example, suppose a method accepts three arguments:

DoSomeThing*namedarg1, namedarg2, namedarg3*

By assigning values to named arguments, you can use the following statement:

DoSomeThing namedarg3 := 4, namedarg2 := 5, namedarg1 := 20

Note that the named arguments don't have to appear in the normal positional order in the syntax.

Null

A value indicating that a variable contains no valid data. **Null** is the result of an explicit assignment of **Null** to a variable or any operation between expressions that contain **Null**.

numeric data type

Any intrinsic numeric data type (**Byte**, **Boolean**, **Integer**, **Long**, **Currency**, **Single**, **Double**, or **Date**).

numeric expression

Any expression that can be evaluated as a number. Elements of an expression can include any combination of keywords, variables, constants, and operators that result in a number.

numeric type

Any intrinsic numeric data type (**Byte**, **Boolean**, **Integer**, **Long**, **Currency**, **Single**, **Double**, or **Date**) or any **Variant** numeric subtype (**Empty**, **Integer**, **Long**, **Single**, **Double**, **Currency**, **Decimal**, **Date**, **Error**, **Boolean**, or **Byte**).

object

A combination of code and data that can be treated as a unit, for example, a control, form, or application component. Each object is defined by a class.

Object box

A list box at the upper-left corner of the **Code** window that lists the form and controls in the form to which the code is attached, or a list box located at the top of the **Properties** window that lists the form and its controls.

Object Browser

A dialog box in which you can examine the contents of an object library to get information about the objects provided.

Object data type

A data type that represents any **Object** reference. **Object** variables are stored as 32-bit (4-byte) addresses that refer to objects.

object expression

An expression that specifies a particular object and can include any of the object's containers. For example, an application can have an **Application** object that contains a **Document** object that contains a **Text** object.

object library

A file with the .olb extension that provides information to Automation controllers (like Visual Basic) about available objects. You can use the **Object Browser** to examine the contents of an object library to get information about the objects provided.

object module

A module that contains code specific to an object, for example, class module, form module, and document module. Object modules contain the code behind their associated objects. The rules for object modules differ from those for standard modules.

object type

A type of object exposed by an application through Automation, for example, **Application**, **File**, **Range**, and **Sheet**. Use the **Object Browser** or refer to the application's documentation for a complete listing of available objects.

object variable

A variable that contains a reference to an object.

Automation object

An object that is exposed to other applications or programming tools through Automation interfaces.

parameter

Variable name by which an argument passed to a procedure is known within the procedure. This variable receives the argument passed into the procedure. Its scope ends when the procedure ends.

path

A string expression specifying a directory or folder location. The location can include a drive specification.

pi

A mathematical constant equal to approximately 3.1415926535897932.

point

A point is 1/72 inch. Font sizes are usually measured in points.

print zone

Print zones begin every 14 columns. The width of each column is an average of the width of all characters in the point size for the selected font.

Private

Variables that are visible only to the module in which they are declared.

procedure

A named sequence of statements executed as a unit. For example, **Function**, **Property**, and **Sub** are types of procedures. A procedure name is always defined at module level. All executable code must be contained in a procedure. Procedures can't be nested within other procedures.

Procedure box

A list box at the upper-right corner of the **Code** window and the **Debug** window that displays the procedures recognized for the object displayed in the **Object** box.

procedure call

A statement in code that tells Visual Basic to execute a procedure.

procedure level

Describes statements located within a **Function**, **Property**, or **Sub** procedure. Declarations are usually listed first, followed by assignments and other executable code.

Note that module-level code resides outside a procedure block.

project

A set of modules.

Project window

A window that displays a list of the form, class, and standard modules; the resource file; and references in your project. Files with .ocx and .vbx file name extensions aren't displayed in the **Project** window.

Properties window

A window used to display or change properties of a selected form or control at design time. Some custom controls have customized **Properties** windows.

property

A named attribute of an object. Properties define object characteristics such as size, color, and screen location, or the state of an object, such as enabled or disabled.

Property procedure

A procedure that creates and manipulates properties for a class module. A **Property** procedure begins with a **Property Let**, **Property Get**, or **Property Set** statement and ends with an **End Property** statement.

Public

Variables declared using the **Public** statement are visible to all procedures in all modules in all applications unless **Option Private Module** is in effect. In that case, the variables are public only within the project in which they reside.

referenced project

The project you directly create a link to from the current project you are working on. A project referenced by one of the current project's directly referenced projects is called an indirectly referenced project. Its **Public** variables are not accessible to the current project except through qualification with its project name. Any combination of direct and indirect references between projects is valid as long as they do not result in a complete cycle.

referencing project

The current project. How you create a link to a project depends on the host application. For example, to directly reference a project in Microsoft Excel, select the project from the **References** dialog box of the **Tools** menu. **Public** variables in a directly referenced project are visible to the directly referencing project, but **Public** variables in a directly referencing project are not visible to a directly referenced project.

registry

In Microsoft Windows version 3.1, OLE registration information and file associations are stored in the registration database, and program settings are stored in Windows system initialization (.ini) files. In Microsoft Windows 95, the Windows registry serves as a central configuration database for user, application, and computer-specific information, including the information previously contained in both the Windows version 3.1 registration database and .ini files.

resource file

A file in a Visual Basic project with an .res file name extension that can contain bitmaps, text strings, or other data. By storing this data in a separate file, you can change the information without editing your code. Only one resource file can be associated with a project.

run-time error

An error that occurs when code is running. A run-time error results when a statement attempts an invalid operation.

run time

The time during which code is running. During run time, you can't edit the code.

scope

Defines the visibility of a variable, procedure, or object. For example, a variable declared as **Public** is visible to all procedures in all modules in a directly referencing project unless **Option Private Module** is in effect. When **Option Private Module** is in effect, the module itself is private and therefore not visible to referencing projects. Variables declared in a procedure are visible only within the procedure and lose their value between calls unless they are declared **Static**.

seed

An initial value used to generate pseudorandom numbers. For example, the **Randomize** statement creates a seed number used by the **Rnd** function to create unique pseudorandom number sequences.

Single data type

A data type that stores single-precision floating-point variables as 32-bit (4-byte) floating-point numbers, ranging in value from -3.402823E38 to -1.401298E-45 for negative values, and 1.401298E-45 to 3.402823E38 for positive values. The exclamation point (**!**) type-declaration character represents a **Single** in Visual Basic.

sort order

A sequencing principle used to order data, for example, alphabetic, numeric, ascending, descending, and so on.

stack

A fixed amount of memory used by Visual Basic to preserve local variables and arguments during procedure calls.

standard module

A module containing only procedure, type, and data declarations and definitions. Module-level declarations and definitions in a standard module are **Public** by default. A standard module is referred to as a code module in earlier versions of Visual Basic.

statement

A syntactically complete unit that expresses one kind of action, declaration, or definition. A statement generally occupies a single line, although you can use a colon (**:**) to include more than one statement on a line. You can also use a line-continuation character (**_**) to continue a single logical line onto a second physical line.

string comparison

A comparison of two sequences of characters. Use **Option Compare** to specify binary or text comparison. In English-U.S., binary comparisons are case sensitive; text comparisons are not.

string constant

Any constant (defined using the **Const** keyword) consisting of a sequence of contiguous characters interpreted as the characters themselves rather than as a numeric value.

String data type

A data type consisting of a sequence of contiguous characters that represent the characters themselves rather than their numeric values. A **String** can include letters, numbers, spaces, and punctuation. The **String** data type can store fixed-length strings ranging in length from 0 to approximately 63K characters and dynamic strings ranging in length from 0 to approximately 2 billion characters. The dollar sign (**$**) type-declaration character represents a **String** in Visual Basic.

string expression

Any expression that evaluates to a sequence of contiguous characters. Elements of a string expression can include a function that returns a string, a string literal, a string constant, a string variable, a string **Variant**, or a function that returns a string **Variant** (**VarType** 8).

string literal

Any expression consisting of a sequence of contiguous characters surrounded by quotation marks that is literally interpreted as the characters within the quotation marks.

Sub procedure

A procedure that performs a specific task within a program, but returns no explicit value. A **Sub** procedure begins with a **Sub** statement and ends with an **End Sub** statement.

syntax checking

A feature that checks your code for correct syntax. If the syntax checking feature is enabled, a message is displayed when you enter code that contains a syntax error and the suspect code is highlighted.

syntax error

An error that occurs when you enter a line of code that Visual Basic doesn't recognize.

Note that syntax rules for individual keywords are defined in the Syntax section of the associated Help topic. To get Help on a keyword from within the development environment, select the keyword and press F1.

tab order

The order in which the focus moves from one field to the next as you press TAB or SHIFT+TAB.

time expression

Any expression that can be interpreted as a time. This includes any combination of time literals, numbers that look like times, strings that look like times, and times returned from functions.

Times are stored as part of a real number. Values to the right of the decimal represent the time. For example, midday (12:00 P.M.) is represented by 0.5.

twip

A unit of screen measurement equal to 1/20 point. A twip is a screen-independent unit used to ensure that placement and proportion of screen elements in your screen application are the same on all display systems. There are approximately 1440 twips to a logical inch or 567 twips to a logical centimeter (the length of a screen item measuring one inch or one centimeter when printed).

type-declaration character

A character appended to a variable name indicating the variable's data type. By default, variables are of type **Variant** unless a corresponding **Def***type* statement is present in the module.

type library

A file or component within another file that contains standard descriptions of exposed objects, properties, and methods that are available for Automation. Object library files (.olb) contain type libraries.

Unicode

International Standards Organization (ISO) character standard. Unicode uses a 16-bit (2-byte) coding scheme that allows for 65,536 distinct character spaces. Unicode includes representations for punctuation marks, mathematical symbols, and dingbats, with substantial room for future expansion.

universal date format

The universal date format is `#yyyy-mm-dd hh:mm:ss#`. However, both the date component (`#yyyy-mm-dd#`) and the time component (`#hh:mm:ss#`) can be represented separately.

user-defined type

Any data type defined using the **Type** statement. User-defined data types can contain one or more elements of any data type. Arrays of user-defined and other data types are created using the **Dim** statement. Arrays of any type can be included within user-defined types.

variable

A named storage location that can contain data that can be modified during program execution. Each variable has a name that uniquely identifies it within its scope. A data type can be specified or not.

Variable names must begin with an alphabetic character, must be unique within the same scope, can't be longer than 255 characters, and can't contain an embedded period or type-declaration character.

Variant data type

A special data type that can contain numeric, string, or date data as well as user-defined types and the special values **Empty** and **Null**. The **Variant** data type has a numeric storage size of 16 bytes and can contain data up to the range of a **Decimal**, or a character storage size of 22 bytes (plus string length), and can store any character text. The **VarType** function defines how the data in a **Variant** is treated. All variables become **Variant** data types if not explicitly declared as some other data type.

variant expression

Any expression that can evaluate to numeric, string, or date data, as well as the special values **Empty** and **Null**.

watch expression

A user-defined expression that enables you to observe the behavior of a variable or expression. Watch expressions appear in the Watch window of the Visual Basic Editor and are automatically updated when you enter break mode. The **Watch** window displays the value of an expression within a given context. Watch expressions are not saved with your code.

z-order

The visual layering of controls on a form along the form's z-axis (depth). The z-order determines which controls are in front of other controls.

Data Types:-

In computer science and computer programming, a data type or simply type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data. Most programming languages support various types of data, for example: real, integer or Boolean. A data type provides a set of values from which an expression (i.e. variable, function...) may take its values. This data type defines the operations that can be done on the data, the meaning of the data, and the way values of that type can be stored. A type of value from which an expression may take its value.

Data:-

Data and information are often used interchangeably; however, the extent to which a set of data is informative to someone depends on the extent to which it is unexpected by that person. The amount of information content in a data stream may be characterized by its Shannon entropy.

While the concept of data is commonly associated with scientific research, data is collected by a huge range of organizations and institutions, including businesses (e.g., sales data, revenue, profits, stock price), governments (e.g., crime rates, unemployment rates, literacy rates) and non-governmental organizations (e.g., censuses of the number of homeless people by non-profit organizations).

Data is measured, collected and reported, and analyzed, whereupon it can be visualized using graphs, images or other analysis tools. Data as a general concept refers to the fact that some existing information or knowledge is *represented* or *coded* in some form suitable for better usage or processing. *Raw data* ("unprocessed data") is a collection of numbers or characters before it has been "cleaned" and corrected by researchers. Raw data needs to be corrected to remove outliers or obvious instrument or data entry errors (e.g., a thermometer reading from an outdoor Arctic location recording a tropical temperature). Data processing commonly occurs by stages, and the "processed data" from one stage may be considered the "raw data" of the next stage. Field data is raw data that is collected in an uncontrolled "in situ" environment. Experimental data is data that is generated within the context of a scientific investigation by observation and recording. Data has been described as the new oil of the digital economy.

Real Number:-

In mathematics, a **real number** is a value of a continuous quantity that can represent a distance along a line. The adjective *real* in this context was introduced in the 17th century by René Descartes, who distinguished between real and imaginary roots of polynomials. The real numbers include all the rational numbers, such as the integer −5 and the fraction 4/3, and all the irrational numbers, such as $\surd\boxed{2}$ (1.41421356..., the square root of 2, an irrational algebraic number). Included within the irrationals are the transcendental numbers, such as π (3.14159265...). In addition to measuring distance, real numbers can be used to measure quantities such as time, mass, energy, velocity, and many more.

Real numbers can be thought of as points on an infinitely long line called the number line or real line, where the points corresponding to integers are equally spaced. Any real number can be determined by a possibly infinite decimal representation, such as that of 8.632, where each consecutive digit is measured in units one tenth the size of the previous one. The real line can be thought of as a part of the complex plane, and complex numbers include real numbers.

Integer:-

In computer science, an **integer** is a datum of **integral data type**, a data type that represents some range of mathematical integers. Integral data types may be of different sizes and may or may not be allowed to contain negative values. Integers are commonly represented in a computer as a group of binary digits (bits). The size of the grouping varies so the set of integer sizes available varies between different types of computers. Computer hardware, including virtual machines, nearly always provide a way to represent a processor register or memory address as an integer.

Boolean:-

In computer science, the **Boolean data type** is a data type, having two values (usually denoted *true* and *false*), intended to represent the truth values of logic and Boolean algebra. It is named after George Boole, who first defined an algebraic system of logic in the mid 19th century. The Boolean data type is primarily associated with conditional statements, which allow different actions and change control flow depending on whether a programmer-specified Boolean *condition* evaluates to true or false. It is a special case of a more general *logical data type*; logic need not always be Boolean.

Variable declaration:-

To create a new variable

1. Declare the variable in a `Dim` statement.

□ Dim new Customer

□ Include specifications for the variable's characteristics, such as Private, Static, Shadows, or With Events. For more information, see Declared Element Characteristics.

□ Public Static new Customer

2. You do not need the `Dim` keyword if you use other keywords in the declaration.

□ Follow the specifications with the variable's name, which must follow Visual Basic rules and conventions. For more information, see Declared Element Names.

□ Public Static newCustomer

□ Follow the name with the As clause to specify the variable's data type.

□ Public Static newCustomer As Customer

3. If you do not specify the data type, it uses the default: `Object`.

□ Follow the `As` clause with an equal sign (=) and follow the equal sign with the variable's initial value.

4. Visual Basic assigns the specified value to the variable every time it runs the Dim statement. If you do not specify an initial value, Visual Basic assigns the default initial value for the variable's data type when it first enters the code that contains the Dim statement.

5. If the variable is a reference type, you can create an instance of its class by including the New Operator keyword in the `As` clause. If you do not use `New`, the initial value of the variable is Nothing.
6. Public Static newCustomer As New Customer .

Array:-

An array is a set of values, which are termed *elements*, that are logically related to each other. For example, an array may consist of the number of students in each grade in a grammar school; each element of the array is the number of students in a single grade. Similarly, an array may consist of a student's grades for a class; each element of the array is a single grade.

It is possible individual variables to store each of our data items. For example, if our application analyzes student grades, we can use a separate variable for each student's grade, such as englishGrade1, englishGrade2, etc. This approach has three major limitations:

- We have to know at design time exactly how many grades we have to handle.
- Handling large numbers of grades quickly becomes unwieldy. This in turn makes an application much more likely to have serious bugs.
- It is difficult to maintain. Each new grade that we add requires that the application be modified, recompiled, and redeployed.

By using an array, you can refer to these related values by the same name, and use a number that's called an *index* or *subscript* to identify an individual element based on its position in the array. The indexes of an array range from 0 to one less than the total number of elements in the array. When you use Visual Basic syntax to define the size of an array, you specify its highest index, not the total number of elements in the array. You can work with the array as a unit, and the ability to iterate its elements frees you from needing to know exactly how many elements it contains at design time.

Some quick examples before explanation:

VB

```
' Declare a single-dimension array of 5 numbers.
Dim numbers(4) As Integer

'Declare a single-dimension array and set its 4 values.
Dim numbers = New Integer() {1, 2, 4, 8}

' Change the size of an existing array to 16 elements and retain the current values.
ReDim Preserve numbers(15)

' Redefine the size of an existing array and reset the values.
ReDim numbers(15)

' Declare a 6 x 6 multidimensional array.
Dim matrix(5, 5) As Double

' Declare a 4 x 3 multidimensional array and set array element values.
Dim matrix = New Integer(3, 2) {{1, 2, 3}, {2, 3, 4}, {3, 4, 5}, {4, 5, 6}}

' Declare a jagged array
Dim sales()() As Double = New Double(11)() {}
```

Internet and Web Page Designing

Introduction:-

Internet And Web page Designing:

In today’s world, it is the rare person who has not had some exposure to the Internet and the World-Wide Web. According to recent research, as of the year 2004 there were only about 20% of American homes without a computer while 50% of homes had some form of high-speed internet connection. Many of us have not only used the Internet but have also created web content in some form or other. The purpose of this chapter is to provide you with a brief introduction to and history of the Internet and the World-Wide Web as well as computer monitor technology and to give you a basic understanding of how they work. While a thorough and technical history is beyond the scope of this work, having some idea of the history and development of the Internet and the web is helpful in understanding the design constraints imposed by the technology.

For example

you may need to know why the text formatting of a web page is limited to a few sizes, basic fonts and type styles, why the resolution of a web photo is so low, why exact placement of content is difficult and why images are linked to rather than embedded in HTML documents. As you study web design, you will come to understand and appreciate why there are so many constraints and how we can work around them (or with them) - Facilitating our efforts to become good web designers. It must be said at the outset that the history of the Internet and the web is not exactly cut and dried. There are many versions of the history and many different people who are given credit for the development and continuation of these technologies. The information presented here is a compilation of many of these stories gleaned from various sources.

Networks:

In order to understand the history and workings of the Internet and the web you first need to understand a little about computer networking. A network in its simplest form is just a series of interconnected people, operations, broadcast stations or computers. One example of a network is the old telegraph system that you may remember seeing in a western movie The telegraph enabled communication between two or more remote locations by using simple binary code (Morse code). A message was first converted from spoken or written language onto a series of clicks with short or long spaces between them (encoding). For example S-O-S, the familiar distress signal is encoded as “... --- ...” On the receiving end, the clicks and spaces could be “decoded” by a trained operator into an intelligible message.When computers were first introduced, it soon became apparent that the ability to share information between them was a valuable and desirable capability. Initially, if someone wanted to share data from one computer with another, the information from the first computer had to be printed out in some manner and manually entered into the second computer. With the advent of paper punch cards, and later, magnetic and optical storage devices, the process of data transfer became easier and fewer errors occurred than when done manually. But if data could be exchanged directly between computers tremendous increases in the transfer time and operator efficiency could be realized. To facilitate this information exchange, simple computer networks, sometimes called “peer-to-peer” networks were developed. These networks basically allow the sharing of information among all the users equally. Each workstation can allow or restrict the sharing of specific files, folders and devices such as printers with other workstations. Where more than two computers are connected in this way, the use of a device called a “hub or switch” facilitates the connections among the various workstations. The computers do not need to be in the same room or even the same city, they can be connected remotely.

Evolution of web design:

In 1996, Microsoft released its first competitive browser, which was complete with its own features and tags. It was also the first browser to support style sheets, which at the time was seen as an obscure authoring technique.[5] The HTML markup for tables was originally intended for displaying tabular data. However designers quickly realized the potential of using HTML tables for creating the complex, multi-column layouts that were otherwise not possible. At this time, as design and good aesthetics seemed to take precedence over good mark-up structure, and little attention was paid to semantics and web accessibility. HTML sites were limited in their design options, even more so with earlier versions of HTML. To create complex designs, many web designers had to use complicated table structures or even use blank spacer .GIF images to stop empty table cells from collapsing.[6] CSS was introduced in December 1996 by the W3C to support presentation and layout. This allowed HTML code to be semantic rather than both semantic and presentational, and improved web accessibility, see tableless web design.

In 1996, Flash (originally known as FutureSplash) was developed. At the time, the Flash content development tool was relatively simple compared to now, using basic layout and drawing tools, a limited precursor to ActionScript, and a timeline, but it enabled web designers to go beyond the point of HTML, animated GIFs and JavaScript. However, because Flash required a plug-in, many web developers avoided using it for fear of limiting their market share due to lack of compatibility. Instead, designers reverted to gif animations (if they didn't forego using motion graphics altogether) and JavaScript for widgets. But the benefits of Flash made it popular enough among specific target markets to eventually work its way to the vast majority of browsers, and powerful enough to be used to develop entire sites.

PROTOCOL:-

A protocol is a rule which describes how an activity should be performed, especially in the field of diplomacy. In diplomatic services and governmental fields of end eavor protocols are often unwritten guidelines. Protocols specify the proper and generally accepted behavior in matters of state and diplomacy, such as showing appropriate respect to a head of state, ranking diplomats in chronological order of their accreditation at court, and so on. One definition is:

Protocol is commonly described as a set of international courtesy rules. These well-established and time-honored rules have made it easier for nations and people to live and work together. Part of protocol has always been the acknowledgment of the hierarchical standing of all present. Protocol rules are based on the principles of civility. Dr. P.M. Forni on behalf of the International Association of Protocol Consultants and Officers.

There are two meanings of the word *protocol*. In the legal sense, it is defined as an international agreement that supplements or amends a treaty. In the diplomatic sense, the term refers to the set of rules, procedures, conventions and ceremonies that relate to relations between states. In general, protocol represents the recognized and generally accepted system of international courtesy.

The term *protocol* is derived, via French and Medieval Latin, from the Greek word πρωτόκολλον *protokollon* "first glued sheet of *or* onto a papyrus-roll". This comes from the act of gluing a sheet of paper to the front of a document to preserve it when it was sealed, which imparted additional authenticity to it. In the beginning, the term protocol related to the various forms of interaction observed in official correspondence between states, which were often elaborate in nature. In course of time, however, it has come to cover a much wider range of international relations.

Interface Concepts:-

Interfaces is a bimonthly peer-reviewed academic journal about operations research that was established by The Institute of Management Sciences, now part of the Institute for Operations Research and the Management Sciences. The journal's distinguishing feature is its case-study style: It offers examples of how operations research theory has been applied in businesses and organizations.

Hardware Interface:

An electrical connector, is an electro-mechanical device used to join electrical terminations and create an electrical circuit. Electrical connectors consist of plugs (male-ended) and jacks (female-ended). The connection may be temporary, as for portable equipment, require a tool for assembly and removal, or serve as a permanent electrical joint between two wires or devices.[1] An adapter can be used to effectively bring together dissimilar connectors.

Hundreds of types of electrical connectors are manufactured for power, signal and control applications. Connectors may join two lengths of flexible copper wire or cable, or connect a wire or cable to an electrical terminal.

In computing, an electrical connector can also be known as a physical interface (compare physical layer in OSI model of networking). Cable glands, known as *cable connectors* in the US, connect wires to devices mechanically rather than electrically and are distinct from quick-disconnects performing the latter.

Audio Interface:

An electrical connector, is an electro-mechanical device used to join electrical terminations and create an electrical circuit. Electrical connectors consist of plugs (male-ended) and jacks (female-ended). The connection may be temporary, as for portable equipment, require a tool for assembly and removal, or serve as a permanent electrical joint between two wires or devices.[1] An adapter can be used to effectively bring together dissimilar connectors.

Hundreds of types of electrical connectors are manufactured for power, signal and control applications. Connectors may join two lengths of flexible copper wire or cable, or connect a wire or cable to an electrical terminal.

In computing, an electrical connector can also be known as a physical interface (compare physical layer in OSI model of networking). Cable glands, known as *cable connectors* in the US, connect wires to devices mechanically rather than electrically and are distinct from quick-disconnects performing the latter.

Network Interface:

A network interface is a system's (software and/or hardware) interface between two pieces of equipment or protocol layers in a computer network.

A network interface will usually have some form of network address. This may consist of a node ID and a port number or may be a unique node ID in its own right.

Network interfaces provide standardized functions such as passing messages, connecting and disconnecting, etc.

Examples:

- **Computer port (hardware),** an interface to other computers or peripherals
- **Network interface controller**, the device a computer uses to connect to a computer network
- **Network interface device**, a demarcation point for a telephone network
- **Network socket**, a software interface to the network
- **Port (computer networking),** a protocol interface to the network.

User Interface:-

The user interface (UI), in the industrial design field of human–computer interaction, is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision-making process. Examples of this broad concept of user interfaces include the interactive aspects of computer operating systems, hand tools, heavy machinery operator controls, and process controls. The design considerations applicable when creating user interfaces are related to or involve such disciplines as ergonomics and psychology.

Generally, the goal of user interface design is to produce a user interface which makes it easy (self-explanatory), efficient, and enjoyable (user-friendly) to operate a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human.

With the increased use of personal computers and the relative decline in societal awareness of heavy machinery,[clarification needed] the term user interface is generally assumed to mean the graphical user interface, while industrial control panel and machinery control design discussions more commonly refer to human-machine interfaces.

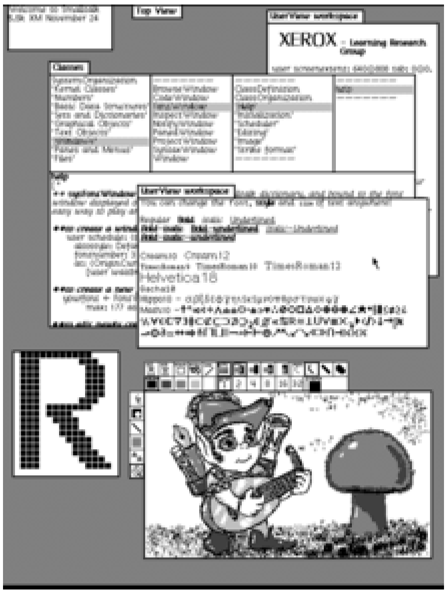
Other terms for user interface are man–machine interface (MMI) and when the machine in question is a computer human–computer interface.

Example of a tangible user interface

Graphical User Interface:-

The graphical user interface (GUI /guːiː/), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs),[which require commands to be typed on a computer keyboard.

The actions in a GUI are usually performed through direct manipulation of the graphical elements. Beyond computers, GUIs are used in many handheld mobile devices such as MP3 players, portable media players, gaming devices, smartphones and smaller household, office and industrial controls. The term GUI tends not to be applied to other lower-display resolution types of interfaces, such as video games (where head-up display (HUD)is preferred), or not including flat screens, like volumetric displays because the term is restricted to the scope of two-dimensional display screens able to describe generic information, in the tradition of the computer science research at the Xerox Palo Alto Research Center.



Example of Graphical user Interface.

Application Programming Interface:

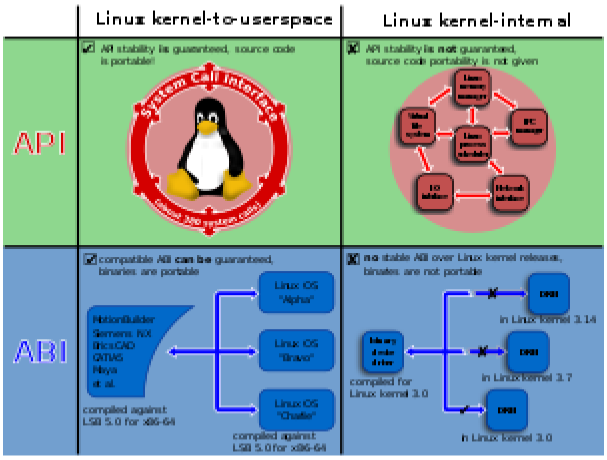
In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, or computer hardware. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables , are examples of different forms of APIs. Documentation for the API is usually provided to facilitate usage and implementation.

Application Binary Interface:-

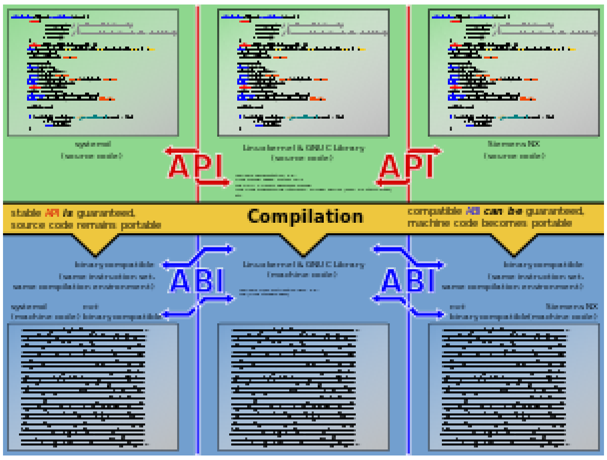
In computer software, an application binary interface (ABI) is an interface between two binary program modules; often, one of these modules is a library or operating system facility, and the other is a program that is being run by a user.

An ABI defines how data structures or computational routines are accessed in machine code, which is a low-level, hardware-dependent format; in contrast, an API defines this access in source code, which is a relatively high-level, relatively hardware-independent, often human-readable format. A common aspect of an ABI is the calling convention, which determines how data is provided as input to or read as output from computational routines; examples are the x86 calling conventions.

Adhering to an ABI (which may or may not be officially standardized) is usually the job of a compiler, operating system, or library author; however, an application programmer may have to deal with an ABI directly when writing a program in a mix of programming languages, which can be achieved by using foreign function calls.


Linux kernel-to-userspace
Linux kernel-internal
API
ABI
API stability is guaranteed, source code is portable!
API stability is not guaranteed, source code portability is not given
System Call Interface
(almost 380 system calls)
compatible ABI can be guaranteed, binaries are portable
no stable ABI over Linux kernel releases, binaries are not portable
Linux OS "Alpha"
Linux OS "Bravo"
Linux OS "Charlie"
MotionBuilder
Siemens NX
BricsCAD
CATIAS
Maya
et al.
compiled against LSB 5.0 for x86-64
compiled against LSB 5.0 for x86-64
compiled for Linux kernel 3.0
in Linux kernel 3.14
in Linux kernel 3.7
in Linux kernel 3.0

A high-level comparison of in-kernel and kernel-to-users pace APIs and ABIs


API
API
Compilation
stable API is guaranteed, source code remains portable
compatible ABI can be guaranteed, machine code becomes portable
ABI
ABI
Linux kernel & GNU C Library (source code)
Linux kernel & GNU C Library (machine code)
Siemens NX (source code)
Siemens NX
binary compatible (same instruction set, same compilation environment)
not binary compatible

Linux kernel and GNU C Library define the Linux API. After compilation, the binaries offer an ABI; keeping this ABI stable over a long time is important for ISVs.

Internet Vs Intranet:-

BASIS FOR COMPARISON	INTERNET	INTRANET
Meaning	Connects different network of computers together	It is a part of Internet which is privately owned by a particular firm
Accessibility	Anyone can access the Internet	Accessible only by the organization members, having login details.
Safety	Is not as safe as compared to Intranet	Safe
No of Users	Unlimited	Limited
Visitors Traffic	More	Less
Network Type	Public	Private
Information Provided	Unlimited, and can be viewed by everyone	Limited, and circulates among the members of an organization

Definition Of Internet:

The Internet is a global network that establishes a connection and provides transmission between various computers. It uses both wired and wireless mode of communication to send and receive any information such as data, audio, video, etc. Here, data travels through "fiber optic cables", which are owned by telephone companies.

Nowadays everyone uses the Internet for acquiring information, communication, and transferring data over the network. It is a public network using which computers can connect and relay to each other. It provides an excellent source of information to the user.

Definition Of Intranet:

An intranet is a part of Internet that is privately owned by an organization. It connects all the computers together and provides access to files and folders within the network. It has a firewall surrounding the system to avoid the unauthorized user from accessing the network. Only authorized users have permission to access the network.

Moreover, Intranet is used for connecting computers and transmitting data, files or documents within the firm. It is a secure way to share the details, materials, and folders as the network are highly secured and restricted within the organization. It renders various services such as email, search, data storage, etc

Key Differences Between Internet and Intranet:

- The Internet provides unlimited information which can be viewed by everyone whereas, in Intranet, data circulates within the organization.
- The Internet provides access to everyone whereas, Intranet permits authenticate users only.
- The Internet is not owned by any single or multiple organization, whereas, Intranet is a private network that belongs to a firm or an institution.
- The Internet is available to all whereas, Intranet is restricted.
- An intranet is safer as compared to the Internet.

Similarities Between Internet and Intranet:

- Both Internet and Intranet can be accessed using any browser.
- They use Internet Protocols for transferring data.
- Both of them are used to share information with the users over the network.

Conclusion:

Hence, we conclude that both Internet and Intranet have some similar aspects and dissimilarities as well. The Internet is a collection of various LAN, MAN, and WAN whereas, Intranet mostly is a LAN, MAN or WAN. An intranet is safer as compared to the Internet since the user login keeps updating at regular intervals and it confines to an organization.

Growth of Internet:

The Internet is the main cause of the recent explosion of activity in optical fiber telecommunications. The high growth rates observed on the Internet, and the popular perception that growth rates were even higher, led to an upsurge in research, development, and investment in telecommunications. The telecom crash of 2000 occurred when investors realized that transmission capacity in place and under construction greatly exceeded actual traffic demand. This chapter discusses the growth of the Internet and compares it with that of other communication services. Internet traffic is growing, approximately doubling each year. There are reasonable arguments that it will continue to grow at this rate for the rest of this decade. If this happens, then in a few years, we may have a rough balance between supply and demand.

Introduction:

Optical fiber communications was initially developed for the voice phone system. The feverish level of activity that we have experienced since the late 1990s, though, was caused primarily by the rapidly rising demand for Internet connectivity. The Internet has been growing at unprecedented rates. Moreover, because it is versatile and penetrates deeply into the economy, it is affecting all of society, and therefore has attracted inordinate amounts of public attention.

The aim of this chapter is to summarize the current state of knowledge about the growth rates of the Internet, with special attention paid to the implications for fiber optic transmission. We also attempt to put the growth rates of the Internet into the proper context by providing comparisons with other communications services.

The overwhelmingly predominant view has been that Internet traffic (as measured in bytes received by customers) doubles every three or four months. Such unprecedented rates (corresponding to traffic increasing by factors of between 8 and 16 each year) did prevail (within the US) during the crucial twoyear period of 1995 and 1996, when the Internet first burst onto the scene as a major new factor with the potential to transform the economy. However, as we pointed out in [CoffmanO1] (written in early 1998, based on data through the end of 1997), by 1997 those growth rates subsided to approximate the doubling of traffic each year that had been experienced in the early 1990s. A more recent study [CoffmanO2] provided much more evidence, and in particular more recent evidence, that traffic has been about doubling each year since 1997. (We use a doubling of traffic each year to refer to growth rates between 70% and 150% per year, with the wide range reflecting the uncertainties in the estimates.)

Other recent observers also found that Internet traffic is about doubling each year. The evidence was always plentiful, and the only thing lacking was the interest in investigating the question. By the year 2000, though, the myth of Internet traffic doubling every three or four months was getting hard to accept. Very simple arithmetic shows that such growth rates, had they been sustained throughout the period from 1995 (when they did hold) to the end of 2000, would have produced absurdly high traffic volumes. For example, at the end of 1994, traffic on the NSFNet backbone, which was well 2 instrumented, came to about 15 TB/month. Had just that traffic grown at 1,500% per year (which is what a doubling every three months corresponds to), by the end of 2000, there would have been about 250,000,000 TB/month of backbone traffic in the U.S. If we assume 150 million Internet users in the U.S., that would produce a data flow of about 5 Mb/s for each user around the clock. The assumption of a doubling of traffic every four months produces traffic volumes which are only slighly less absurd.

Growth rates of other communication services:

Telecommunications has been a growth industry for centuries, but growth rates have generally been modest, except for a few episodes, such as the beginnings of the electric telegraph (cf. [Odlyzko3]). For example, the number of pieces of mail delivered in the U.S. grew by a factor of over 50,000 between 1800 and 2000, but that was a growth rate of about 5.6% per year. (If we adjust for population increase, we find a growth rate of about 3.5% in the mail volume per capita.)

The number of phone calls in the U.S. grew by a factor of over 230 between 1900 and 2000, for a compound annual growth rate of 5.6%. (The per capita growth rate was 4.2% during this period.) Long distance calls grew faster, about 12% per year between 1930 and 2000, and transatlantic calls faster yet. (There was just one voice circuit between the U.S. and Europe in 1927, when service was inaugurated. It used radio to span the ocean. 6 This single low quality link grew to 23,000 voice circuits to Western Europe by 1995, for a compound annual growth rate of capacity of 16%.)

One communications industry that has been growing very rapidly recently is wireless communication. Table 2.2 shows the growth of the U.S. cell phone industry, with the number of subscribers as of June of each year, and the revenue figures obtained by doubling those of the first six months of each year (and thus seriously understating the full-year figure). In many other countries, wireless communication has developed faster and plays a bigger role than it does in the U.S.. Still, even in the U.S., at the end of 2000, there were close to 100 million cell phones in use, and the rate of growth was far higher than for traditional wired voice services.

Internet History:

The past 5-10 years have witnessed not only an explosion of activity, but the creation of entirely new sectors within the optical industry. As the concept of WDM began to emerge, many new companies developing WDM transport equipment came into existence. The newer enterprises pushed the older established equipment vendors to more aggressive deployment schedules and a constant downward trend for the corresponding prices of WDM transport equipment followed. In what appeared to be an almost insatiable demand for more bandwidth, a situation arose that allowed the creation of the new companies and the accompanying innovation. Not only did new equipment vendors emerge, but also new national scale carriers were created. This trend is continuing as the concept of optical layering/networking is gaining acceptance and new optical equipment companies are being formed on a regular basis. They deal not only with "traditional" WDM transport equipment, but also with terrestrial ultra long haul systems, regional and metro optimized systems, and various incarnations of optical cross connects.

There were hundreds of developments and contributions enabling this burst of activity. Many of the technical innovations are described in this book and its predecessors. However, perhaps the greatest single factor that fueled this phenomena was the belief and perception that traffic and hence needed capacity were growing at explosive rates. This is a remarkable fact, especially when one recalls that around 1990, both the traditional carriers and most of their equipment vendors still expected the traffic demands to not vary much from the voice demand growths (which historically was around 10% per year). In fact both carriers and equipment vendors were arguing that WDM would not be needed and that going to individual channel rates of at most 10 Gb/s would be adequate. Also, around 1995, the conventional wisdom was that 8 channel WDM systems would suffice well into the foreseeable future. Now it almost appears as if the pendulum has swung the other way. Is too much capacity being deployed and are many of the reported traffic growth rates correct, and if so will they continue? As we explained in the previous section, the early skepticism about the need for high capacity optical transport was rooted in the reality of the telecommunications networks. Up until 1990, they were dominated by voice, which was growing slowly.

Then, by the mid-1990s, they became to be dominated (in terms of capacity) by private lines, which were growing three or four times as fast. And then, in the late 1990s, they came to be dominated by the Internet, which was growing faster still. Before we go through the analyses for the traffic growth on the Internet we must first at least define the Internet and describe the history and structure of it. This is paramount in helping put much of later 12 described growth analyses into perspective. When one now speaks of the Internet, it is usually described as an evolution from ARPANET to NSFNET, and finally to the commercial Internet that now exists. Arguably, the phenomenal growth of the Internet started in 1986 (more than 17 years after its "birth") with NSFNet. However, the path was very complicated and full of many twists and turns in its roughly 40 year history [Cerf, Hobbes, Leiner]. From the very early research in packet switching, academia, industry, and the US government have been intertwined as partners.

Ironically, the beginnings of the Internet can trace itself back to the Cold War and specifically to the launch of Sputnik in 1957. The US government formed the Advanced Research Project Agency (ARPA - the name was later changed to DARPA, Defense Advanced Research Project Agency, and later back to ARPA) the year after the launch with the stated goal of establishing a US lead in technology and science (with emphasis on applications for the military). As ARPA was establishing itself, there were several pivotal works [Klein1, Baran] in the early 1960s on packet switching and computer communications. These works and the efforts they spawned laid many of the foundations that enabled the deployment of distributed packet networks. J.C.R. Licklider (of MIT) [LickC] wrote a series of papers in 1962 in which he “envisioned a globally interconnected array of computers which would enable ‘everything’ to easily access data and programs from any of the sites”. Generically speaking, this idea is not much different from what today’s Internet has become. Of importance is the fact the Licklider was the first head of the computer research program at DARPA (beginning in 1962), and in this role he was instrumental in pushing his concept of networks. Kleinrock published both the first paper on packet switching and the first book on the subject. In addition, Kleinrock convinced several key players of the theoretical feasibility of using packets instead of circuits from communications.

One such person was Larry Roberts, one of the initial architects for the ARPANET. In the 1965-66 time frame ARPA sponsored studies on “cooperative network of [users] sharing computers”[Leiner], and the first ARPANET plans were begun, with the first design papers on ARPANET being published in 1967. Concurrently the National Physical Laboratory (NPL) in England deployed an experimental network called the NPL Network making use of packet switching. It utilized 768 kb/s lines. A year before the Moon landing, in 1968, the first ARPANET requests for proposals were sent out, and the first ARPANET contracts were awarded. Two of the earliest contracts went to UCLA to develop the Network Measurement Center, and to Bolt, Beranek and Newman (BBN) for the Packet Switch contract (to construct the Interface Message Processors or IMPs - effectively the routers). Kleinrock headed the Network Measurement Center at UCLA and it was selected as the first node 13 on the ARPANET.

The first IMP was installed at UCLA and the first host computer was connected in September of 1969. The second node was at Stanford Research Institution (SRI). Two other nodes were added at UCSB and in Utah, so that by the second half of 1969, just months past the first moon landing, the initial four node ARPANET became functional. This was truly the initial ARPANET, and thus a case can be made that this was when the Internet was born. The first message carried over the network went from Kleinrock’s lab to SRI. Supposedly the first packet sent over ARPANET was sent by Charley Kline and as he was trying to log in the system crashed as the letter “G” of “LOGIN” was entered.

One of the next major innovations for the fledgling Internet (i.e., ARPANET) was the introduction of the first host-to-host protocol called Network Control Protocal or NCP, which was first used in ARPANET in 1970. By 1972 all of the ARPANET sites had finished implementing NCP. Hence the users of ARPANET could finally begin to focus on the development of applications - another paramount driver for the phenomenal growth and sustained growth of the internet. It was also in 1970 that the first cross-country link was established for ARPANET by AT&T between UCLA and BBN (at the blinding rate of 56 kb/s). By 1971, the ARPANET had grown to 15 nodes and had 23 hosts. However, perhaps the most influential work that year was the creation of an email program that could send messages across a distributed network. (Email was not among the original design criteria for the ARPANET, and its success caught the creators of this network by surprise.) Ray Tomlinson of BBN developed this, and his original program was based on 2 previous ones [Hobbes]. Tomlinson modified his program for ARPANET in 1972, and at that point its popularity quickly soared. In fact it was at this time that the symbol “@” was chosen.

Arguably Internet email as we know it today can trace its origins directly to this work. Internet email was clearly one of key drivers for the popularity (and hence the phenomenal traffic growth demands) of the Internet and was the first "killer app" for the Net. It was every bit as critical to the Internet's "success" as the spreadsheet applications were to the popularization of the PC. Internet email provided a new model of how people could communication with each other and alter the very nature of collaborations.

Although there was already considerable work being done on packet networks outside the US, the first international connections to the ARPANET (to England via Norway) took place in 1973. To put the time frame in perspective this was the same year that Robert Metcalfe did his PhD which described his idea for Ethernet. Also during this year the number of ARPANET "users" was estimated to be 2000 and that 75% of all the ARPANET traffic (in terms of bytes) was email. One needs to note that in only 1-2 years from its introduction onto the Internet email became the predominant type traffic. The same 14 behavior took place several years later for html (i.e., Web traffic), and to a somewhat lesser degree, this was seen for Napster-like traffic within many networks a few years later. Several other key developments began to take place in the mid 1970s. The initial design specification for TCP published by Vint Cerf and Bob Kahn in 1974 [CerfK].

The NCP protocol which was being utilized at the time, tended to act like a device driver, whereas the future TCP (later TCP/IP) would be much more like a communications protocol. As is discussed later, the evolution from ARPANET's NCP protocol to TCP (which in 1978 was split into TCP and IP) was critical in allowing the future growth and scalability of today's Internet. DARPA had three contracts to implement TCP/IP (at the time still called TCP), at Stanford (led by Cerf), BBN (led by Ray Tomlinson) and UCLA (led by Kirsten). Stanford produced the detailed specification and within a year there were 3 independent implementations of TCP that could interoperate.

It is noted that the basic reasons that led to the separation of TCP (which guaranteed reliable delivery) from IP actually came out of work that was done trying to encode and transport voice through a packet switch. It was found that a tremendous amount of buffering was needed, in order to allow for the appropriate reassembly after transmission was completed. This in turn led to trying to find a way to deliver the packets without requiring a guaranteed level of reliability. In essence, the UDP (User Datagram Protocol) was created to allow users to make use of IP. In addition, it was also in 1978 that the first commercial version of ARPANET came into existence as BBN opened Telenet. In 1981-82 the first plans were being made to "migrate" from NCP to TCP. It is claimed by some that it was this event (TCP was established as THE protocol suite for ARPANET) was truly the birth of the Internet - defined as a connected set of networks, specifically those with TCP/IP. A few years later (in 1983) another major development occurred, which later enabled the Internet to scale with the "explosive" growth and popularity of the future Internet. This was the development of the name server (which evolved into the DNS) [Cerf, Leiner]. The name server was developed at the University of Wisconsin [Hobbes] This made it easy for people to use the network since hosts were assigned names and it was not necessary to remember numeric addresses. Much of the credit for the invention of the DNS (domain name server) is credited to Paul Mockapetris of USC/ISI [Cerf]. The year 1983 was also the date for two other key developments on ARPANET. The first one was the cutover from NCP to TCP on the ARPANET. Secondly, ARPANET was split into ARPANET and MILNET.

Although the road was convoluted, this split was one of the key bifurcations points that later allowed NSFNET to come into existence. Soon thereafter (in 1984) the number of hosts on the ARPANET had grown to 1000, and the next year in 1985 the first registered domain was assigned in 15 March. In 1985 NSFNET was created with a backbone speed of 56 kb/s. Initially there were 5 supercomputing centers that were interconnected. One of the paramount benefits of this was that it allowed an explosion of connections (most importantly from universities) to take place. Two years later in 1987, NSF agreed to work with MERIT Network to manage the NSFNet backbone. The next year (1988) the process of upgrading the NSFNet backbone to one based on T1 (i.e., 1.5 Mb/s links) was begun. In 1987 the number of hosts on the Internet broke the 10,000 number. Two year later in 1989 this had grown to around 100,000, and 3 years after that in 1992 it reached the 1,000,000 value.

It is noted that if you look at how the number of hosts had been growing from 1984 to 1992 that it was still pretty much tracking a growth curve that was LESS than tripling each year (i.e., doubling every 9 months). In the 1985-86 time frame key decision was made that had very long term impact: that TCP/IP would be mandatory for the NSFNet program. In the 1988-1990 time frame a conscious decision was made to connect the Internet to electronic mail carriers, and by 1992 most of the commercial email carriers in the US were “like the Internet”. This was still another development that cemented email as the single most important application to take advantage of the Internet. In 1990 the ARPANET ceased to exist, and arguably NSFNet was the essence of the Internet. The following year Commercial Internet Service Providers began to emerge (PSI, ANS, Sprint Link, to name a few) and the Commercial Internet Xchange (CIX) was organized in 1991 by commercial ISPs to provided transfer points for traffic. NSF’s lifting the restriction on the commercial use of the Net was again one of the pivotal decisions. This was again a key bifurcation point, in that this helped set the stage for the complete commercialization of the Net that would follow only a few years later.

In 1991 the upgrading of the NSFNet backbone continued as the work to upgrade to a T3 (i.e., 45 Mb/s links) began. It also interesting to note that it was the next year (1992) than the term “surfing the Internet” was first coined by Jean Armour Polly [Polly], only two years before the ARPAnet/Internet celebrated its 25th anniversary. It was in the 1993-1995 time period that several major events seemed to emerge which fueled an almost explosive growth in the popularity of the Internet. One of the key ones was the introduction of “browsers” most notable Mosaic. This led to the creation of Netscape that went public in 1995. Even as early as 1994 WWW (i.e., predominantly html) traffic was increasing in volume on the Net. By then it was the second most popular type of traffic, surpassed only by ftp traffic. However, in 1995 WWW traffic surpassed ftp as the greatest amount of traffic. In addition the traditional online dial up systems 16 such as AOL, Prodigy and Compuserve began to provide Internet access. In 1996 the net truly became public with the NSFNet being phased out. Soon thereafter major infrastructure improvements were made within the transport part of the

Internet. The Internet began to upgrade much of its backbone to OC3-OC12 (up to 622 Mb/s) links, and in 1999 upgrades began for much of the Net to OC-48 (2.5 Gb/s) links. 5.

The many Internet growth rates The Internet is very hard to describe. By comparison, even the voice phone system, which is a huge enterprise, far larger in terms of revenues than the Internet, is much simpler. In the phone system, the basic service is well defined and simple to describe. The users have only limited ability to interact with the system. The Internet is completely different. Users interact with the system in a multiplicity of ways, on wildy different time scales, and there are many complicated feedback loops.

The paper [FloydP] is an excellent overview of the problems that arise in attempting to simulate the Internet. The problems of measuring the Internet are also formidable. There are many different measures that are relevant. In this chapter, just as in the papers [CoffmanO1, CoffmanO2], we will concentrate on traffic as measured in bytes. For the optical fiber telecommunications industry, it is capacity that is most relevant. Unfortunately there are numerous problems in measuring capacity. Much of the fiber is not lit, and even when it is lit, often only a few wavelengths are lit. Finally, much of potential capacity is used for restoration, through SONET or other methods. In addition, even at the levels of links used for providing IP traffic, it is hard to obtain accurate capacity measurements, since few carriers provide detailed data.

Further, this type of capacity has a tendency to jump suddenly, as bandwidth is usually increased in large steps (such as going from OC3 to OC12, and then OC48, a phenomenon that contributes to the low utilization of data links [Odlyzko1]). Thus there is little regularity in capacity growth figures. On the other hand, we do find astonoshing regularity in traffic growth, which leads us to propose that a form of “Moore’s Law” applies.

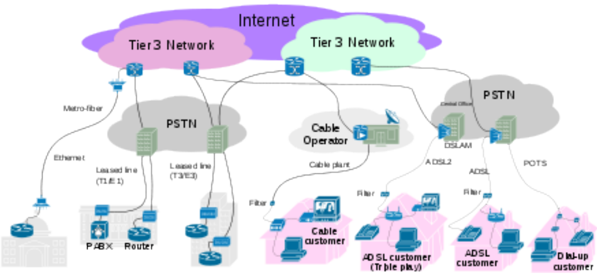
Internet traffic and bandwidth growth:

Whether Internet traffic doubles every three months or just once a year has huge consequences for network design as well as the telecommunications equipment industry. Much of the excitement about and funding for novel technologies appear to be based on expectations of unrealistically high growth rates ([Bruno]). In this section we briefly examine a variety of examples in an attempt to understand the traffic growth rates that the Internet has experienced over its lifetime. There are places where the traffic is growing at rates that exceed 100% per year. One such example is LINX (London Internet Exchange). Its online data, available at http://ochre.linx.net/ , clearly shows a growth rate of about 300% from early 1999 to early 2001. There are also examples with growth rates even higher, although those tend to be for much smaller links or exchange points. However, there are also numerous examples of much more slowly growing links. In this section we briefly present growth rates from a variety of sources, and attempt to put them into context. In an earlier study [CoffmanO1] in 1997 we found that the evidence supported a traffic growth rate of about 100% per year (doubling annually). Four years later, the general conclusion is that Internet traffic still appears to be growing at about 100% per year. In other words, we have not found any substantial slowdown in the growth rate.

Some recent reports and projections conclude that Internet traffic is only about doubling each year, but claim that it was growing much faster until recently, and that its growth rate will continue to slow down. In that view, the telecom crash of 2000 was associated with a sudden decline in the growth rate of traffic. As far as we can tell, that is not accurate. The general rate of growth of traffic appears to have been remarkably stable throughout the period 1997-2000. As one of the most convincing pieces confirming this claim, we cite the news story [Cochrane], based on official figures from Telstra, the dominant Australian telecommunications carrier. This story reports that Telstra's IP traffic was almost exactly doubling each year between November 1997 and November 2000. (The printed version of this news story, but not the one available online at the URL listed in [Cochrane], shows a very regular growth, about 100% per year, from the beginning of 1997 to November 2000.) Hence our conclusion is that the problem the photonics industry is experiencing are not caused by any sudden slowdown in traffic, but rather by a realization that the astronomical growth rates that people had been assuming were phantasies.

Most of this section is drawn from the more detailed account in [CoffmanO2]. There are only a few new pieces of information. For example, the China Internet Network Information Center has statistics (at www.cnnic.net.cn/develst/e-index.shtml) of the Internet bandwidth between China and the rest of the world. It grew from 84.64 Mb/s in June 1998 to 2,799 Mb/s in December 2000, for a compund growth rate of 305% per year. Thus even in a rapidly growing economy like that of China, where the Internet penetration is low, and which is trying to catch up with the industrialized world, traffic is only doubling about every six months.

Internet Service Provider:



Internet connectivity options from end-user to tier 3/2 ISPs

An Internet service provider (ISP) is an organization that provides services for accessing, using, or participating in the Internet. Internet service providers may be organized in various forms, such as commercial, community-owned, non-profit, or otherwise privately owned.

Internet services typically provided by ISPs include Internet access, Internet transit, domain name registration, web hosting, Usenet service, and colocation.

The term *web service* is either

- (generic) a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or

- (specific) a web service implemented in the particular technology or brand, W3C Web Services.

In a web service, the Web technology such as HTTP—originally designed for human-to-machine communication—is utilized for machine-to-machine communication, more specifically for transferring machine-readable file formats such as XML and JSON.

In practice, a web service commonly provides an object-oriented web-based interface to a database server, utilized for example by another web server, or by a mobile app, that provides a user interface to the end user. Many organizations that provide data in formatted HTML pages will also provide that data on their server as XML or JSON, often through a web service to allow syndication, for example Wikipedia's Export. Another application offered to the end user may be a mashup, where a web server consumes several web services at different machines, and compiles the content into one user interface.

Connectivity :-

- Dial-up
- Leased line
- VSAT

Dial – Up Connectivity:-

- **Dial-up Internet access** is a form of Internet access that uses the facilities of the public switched telephone network (PSTN) to establish a connection to an Internet service provider (ISP) by dialing a telephone number on a conventional telephone line. The user's computer or router uses an attached modem to encode and decode information into and from audio frequency signals, respectively.
- In 1979, Tom Truscott and Steve Bellovin, graduate students for Duke University, created an early predecessor to dial-up Internet access called the USENET. The USENET was a UNIX based system that used a dial-up connection to transfer data through telephone modems.[1] Dial-up Internet has been around since the 1980s via public providers such as NSFNET-linked universities and was first offered commercially in July 1992 by Sprint.[2] Despite losing ground to broadband since the mid-2000s, dial-up may still be used where other forms are not available or the cost is too high, such as in some rural or remote areas.

A **modem** (**mo**dulator–**dem**odulator) is a network hardware device that modulates one or more carrier wavesignals to encode digital information for transmission and demodulates signals to decode the transmitted information. The goal is to produce a signal that can be transmitted easily and decoded to reproduce the original digital data. Modems can be used with any means of transmitting analog signals, from light-emitting diodes to radio. A common type of modem is one that turns the digital data of a computer into modulated electrical signal for transmission over telephone lines and demodulated by another modem at the receiver side to recover the digital data.

Modems are generally classified by the maximum amount of data they can send in a given unit of time, usually expressed in bits per second (symbol **bit(s)**, sometimes abbreviated "bps"), or bytes per second (symbol **B(s)**). Modems can also be classified by their symbol rate, measured in baud. The baud unit denotes symbols per second, or the number of times per second the modem sends a new signal. For example, the ITU V.21 standard used audio frequency-shift keying with two possible frequencies, corresponding to two distinct symbols (or one bit per symbol), to carry 300 bits per second using 300 baud. By contrast, the original ITU V.22 standard, which could transmit and receive four distinct symbols (two bits per symbol), transmitted 1,200 bits by sending 600 symbols per second (600 baud) using phase-shift keying.

Leased line connectivity:

A leased line is a private bidirectional or symmetric telecommunications circuit between two or more locations provided in exchange for a monthly rent. Sometimes known as a private circuit or data line in the UK.

Unlike traditional PSTN lines they do not have telephone numbers, each side of the line being permanently connected and dedicated to the other. Leased lines can be used for telephone, Internet, or other data services. Some are ringdown services, and some connect to a private branch exchange or router.

Typically, leased lines are used by businesses to connect geographically distant offices. Unlike dial-up connections, a leased line is always active. The fee for the connection is a fixed monthly rate. The primary factors affecting the monthly fee are distance between end points and the speed of the circuit. Because the connection does not carry anybody else's communications, the carrier can assure a given level of quality.

An Internet leased line is a premium Internet connectivity product, normally delivered over fiber, which provides uncontended, symmetrical speeds with full duplex. It is also known as an ethernet leased line, dedicated line, data circuit or private line.

For example, a T1 can be leased and provides a maximum transmission speed of 1.544 Mbit/s. The user can channelize the T1 to separate the 24 DS0 circuits for voice communication, partial the T1 for data and voice communications, or multiplex the channels into a single data circuit. Leased lines, as opposed to DSL, are being used by companies and individuals for Internet access because they afford faster data transfer rates and are cost-effective for heavy users of the Internet.

VSAT (Very Small Aperture Terminal) Connectivity:

VSAT (Very Small Aperture Terminal) is a satellite communications system that serves home and business users. A VSAT end user needs a box that interfaces between the user's computer and an outside antenna with a transceiver. The tranceiver receives or sends a signal to a satellite transponder in the sky. The satellite sends and receives signals from an earth station computer that acts as a hub for the system. Each end user is interconnected with the hub station via the satellite in a star topology. For one end user to communicate with another, each transmission has to first go to the hub station which retransmits it via the satellite to the other end user's VSAT. VSAT handles data, voice, and video signals.

Concepts:-

- POP
- WEB Based E-mail
- Merits
- Address
- Basic of Sending & Receiving
- E-mail Protocols
- Mailing List
- Free E-mail services
- FTP.

POP(point-to-point connection):-

A **point-to-point** connection refers to a communications connection between two Communication endpoints or nodes. An example is a telephone call, in which one telephone is connected with one other, and what is said by one caller can only be heard by the other. This is contrasted with a *point-to-multipoint* or *broadcast* connection, in which many nodes can receive information transmitted by one node. Other examples of point-to-point communications links are leased lines, microwave radio relay and two-way radio.

A point-to-point wireless unit with built-in antenna at Huntington Beach, California.

The term is also used in computer networking and computer architecture to refer to a wire or other connection that links only two computers or circuits, as opposed to other network topologies such as buses or crossbar switches which can connect many communications devices. Point-to-point is sometimes abbreviated as P2P. This usage of P2P is distinct from P2P referring to peer-to-peer for file sharing networks.

WEB Based E-mail:-

Webmail (or web-based email) is any email client implemented as a web application running on a web server. Examples of webmail software are Round cube and SquirrelMail. Examples of webmail providers are **AOL Mail, Gmail, Outlook.com/Hotmail.com and Yahoo! Mail**. Many webmail providers also offer email access by a desktop email client using standard email protocols, while many internet service providers provide a webmail client as part of the email service included in their internet service package.

As with any web application, webmail's main advantage over the use of a desktop email client is the ability to send and receive email anywhere from a web browser. Its main disadvantage is the need to be connected to the Internet while using it. Other software tools also exist to integrate parts of webmail functionality into an OS (e.g. creating messages directly from third party applications via MAPI).

An **email client**, **email reader** or more formally **mail user agent** (MUA) is a computer program in the category of groupware environments used to access and manage a user's email.

Client is meant to be a role. For example, a web application which provides message management, composition, and reception functions may internally act as an email client; as a whole, it is commonly referred to as webmail. Likewise, email client may be referred to a piece of computer hardware or software whose primary or most visible role is to work as an email client.

A **web application** or **web app** is a client–server computer program which the client (including the user interface and client-side logic) runs in a web browser.[1] Common web applications include webmail, online retail sales, online auctions, wikis, instant messaging services and many other functions.

Web server refers to server software, or hardware dedicated to running said software, that can serve contents to the World Wide Web. A web server processes incoming network requests over the HTTP protocol (and several other related protocols).

MERITS OF INTERNET:

It allows someone to be on one computer system while doing work on another. Information retrieval is the second basic Internet function. Many use of the Internet to locate and download some of the free, quality computer software that has been made available by developers on computers all over the world.

File transfer protocol (FTP) is used to access a remote computer and retrieve files from it. FTP is a quick and easy method if you know the remote computer site where the file is stored.

Once a file has been located, FTP makes transfer of the file to your own computer very easy. There are hosts of directories that have been made accessible for FTP to search for files and businesses can use them to locate files they require for their operations.

Like FTP, Gophers are another information retrieval tools that can be used to access files. A gopher is a computer client tool that enables the user to locate information stored on Internet gopher servers through a series of easy-to-use, hierarchical menus.

Most files and digital information that are accessible through FTP also are available through gophers. The Internet has thousands of gopher servers" sites throughout the world. What makes gopher sites easy to use is that each gopher site contains its own system of menus listing subject-matter topics, local files, and other relevant gopher sites.

There are also educational resources on the Internet. They are in various forms such as journals and database on various aspects of knowledge. For example there are sites where one can access online journals, or learn English.

These sites are of tremendous help to those who are doing academic research. There are also special homepages on special topics or subjects of interest. At the moment, not many people use the Internet for this purpose. It is time we encourage our students to visit these educational sites, especially the English language ones and this will go a long way to improve their level of English.

Commerce on the Internet is already a reality. The communication facilities which are on offer have rapidly become integrated as core business tools. Thus most of the business functions are communicative in nature. The emphasis to date has been on use of the Internet for communications with customers and other companies operating on collaborative ventures.

However, an increasing number are concentrating on transactions between businesses and on-line sales. The Internet Mall companies offer everything from books to flowers to travel. Trading partners can directly communicate with each other without passing through middlemen and inefficient multilayered procedures.

The Internet provides a public and universally available set of technologies for these purposes. Therefore, the Internet is rapidly becoming the technology of choice for electronic commerce because it offers businesses an even easier way to link with other businesses and individuals at a very low cost Handling transactions electronically can reduce transaction costs and delivery time for some goods, especially those that are purely digital (such as software, text Products, images, or videos Marketing Communication is another business function of the Internet.

As the term implies, marketing communication involves the use of communication media to market a product, goods and services. The nature of marketing communication at present is different from the mass communication paradigm of old. In the present era, marketing communication is more likely to involve expenditure on direct response marketing than involve the mass communication paradigm of old.

With the advent of the Internet, this traditional view of communication media has been altered. What we have now is the new many - to - many marketing communication model defining the web or Internet. This offers a radical departure from traditional marketing environments.

DEMERITS OF INTERNET

Communication via the Web

Although there are many advantages, researchers argue some disadvantages of an internet based society. Most of these drawbacks are a result of decreased face-to-face communication and the ability to escape identity.

Negative effects on family communication

Although there are conflicting research findings on this topic, an article published by Science Daily reported that time spent on the Internet was associated with later declines in within house family communication and a decrease in the number of friends and acquaintances with which they kept ties.

Lack of conflict resolution

The same article suggests that communication on the web is taking away necessary human conflict. For example, if an Internet conversation starts getting heated or if someone gets angry, it is too easy to just sign off and not deal with the issue. Resolving conflict is a part of life that shouldn’t be avoided and is usually psychologically helpful.

Lonliness

Another proposed disadvantage is that a couple of studies have found strong correlations between frequency of Internet use and loneliness. People that use the Internet often are more likely to become lonely and depressed than those that don’t. One proposal as to why this is the case is because of a potential reduced social support system as a result of the Internet.

Internet addiction

There is some controversy over whether it is possible to actually be addicted to the Internet or not. Some researchers, like John M. Grohol, claim that it is simply people trying to escape their problems in an online world and cannot be classified as an addiction. He demonstrates his theory with this model and states that what some people call an addiction is just someone that is caught in stage one.

> Other psychologists, including Jennifer R. Ferris, believe that Internet addiction is a true psychological disorder with definable symptoms. The symptoms are comparable to any addiction, withdrawal, loss of relationships or job and significant time consumption.
>
> If an actual addiction exists or doesn't exist, the underlying themes that support the addiction theory are still an issue. Whether people are trying to escape problems and reality or they will go through withdrawal if they aren't surfing the net or chatting, it is still psychologically unhealthy.

Email is a great way of communicating with friends and family at the touch of a button, no matter where in the world you are:

Read on to learn:

- how to begin composing a new email
- where to add recipients to your email
- where to input your subject line
- how to send your email or save it as a draft, for later
- The following steps show you how to send an email using a Gmail account. However, many email accounts or applications follow a similar process for creating and sending a new message.

You'll need:

- a computer with active internet connection

- an email account set up and ready to send and receive emails.

Follow these step-by-step instructions to send an email

Step 1: Log in to your Gmail account so that you are on the dashboard (main page) of your mail account.

Step 2: Click **Compose**.

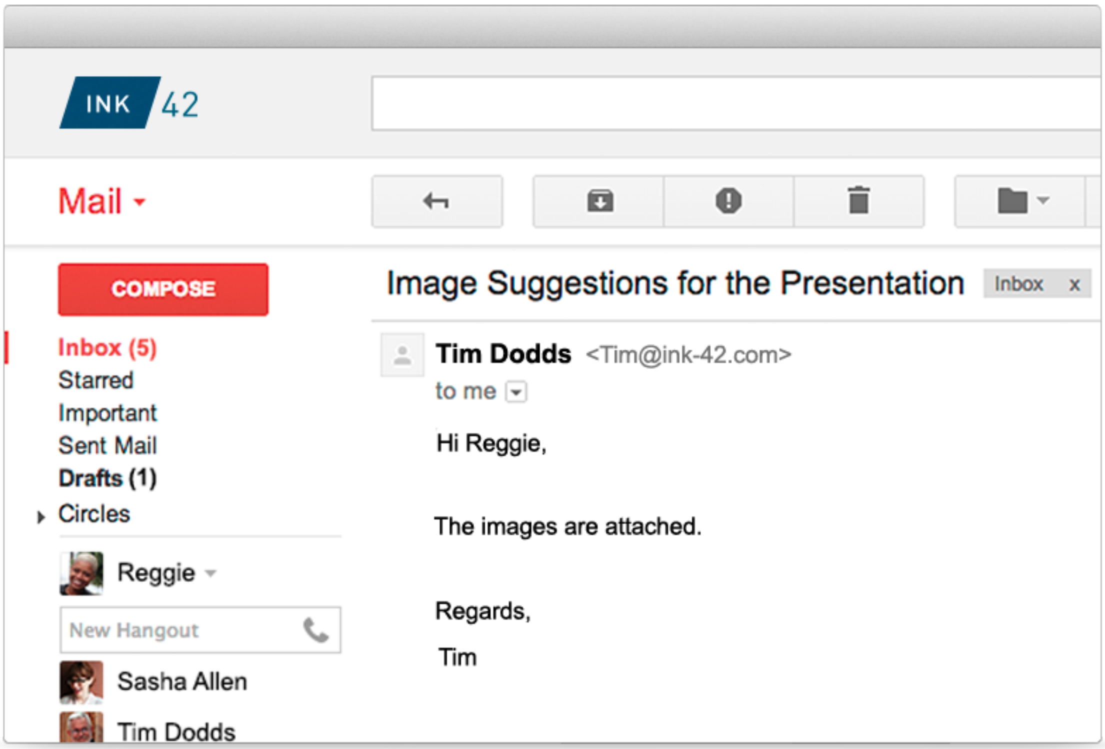


Image Suggestions for the Presentation

Tim Dodds <Tim@ink-42.com>

to me

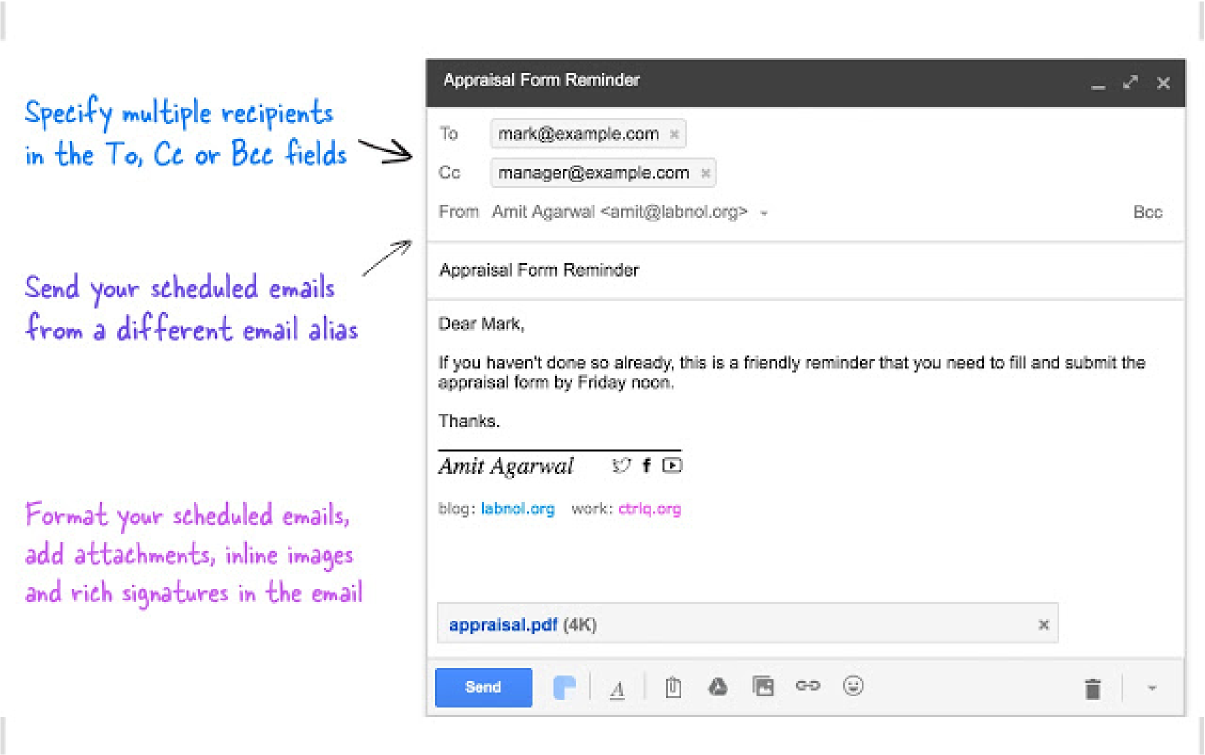
Hi Reggie,

The images are attached.

Regards,

Tim

Step 3: A new blank email window will open up. In the ‘To’ box, type in the email address of the recipient.



Specify multiple recipients in the To, Cc or Bcc fields

Send your scheduled emails from a different email alias

Format your scheduled emails, add attachments, inline images and rich signatures in the email

Appraisal Form Reminder

To mark@example.com

Cc manager@example.com

From Amit Agarwal <amit@labnol.org>

Appraisal Form Reminder

Dear Mark,

If you haven't done so already, this is a friendly reminder that you need to fill and submit the appraisal form by Friday noon.

Thanks.

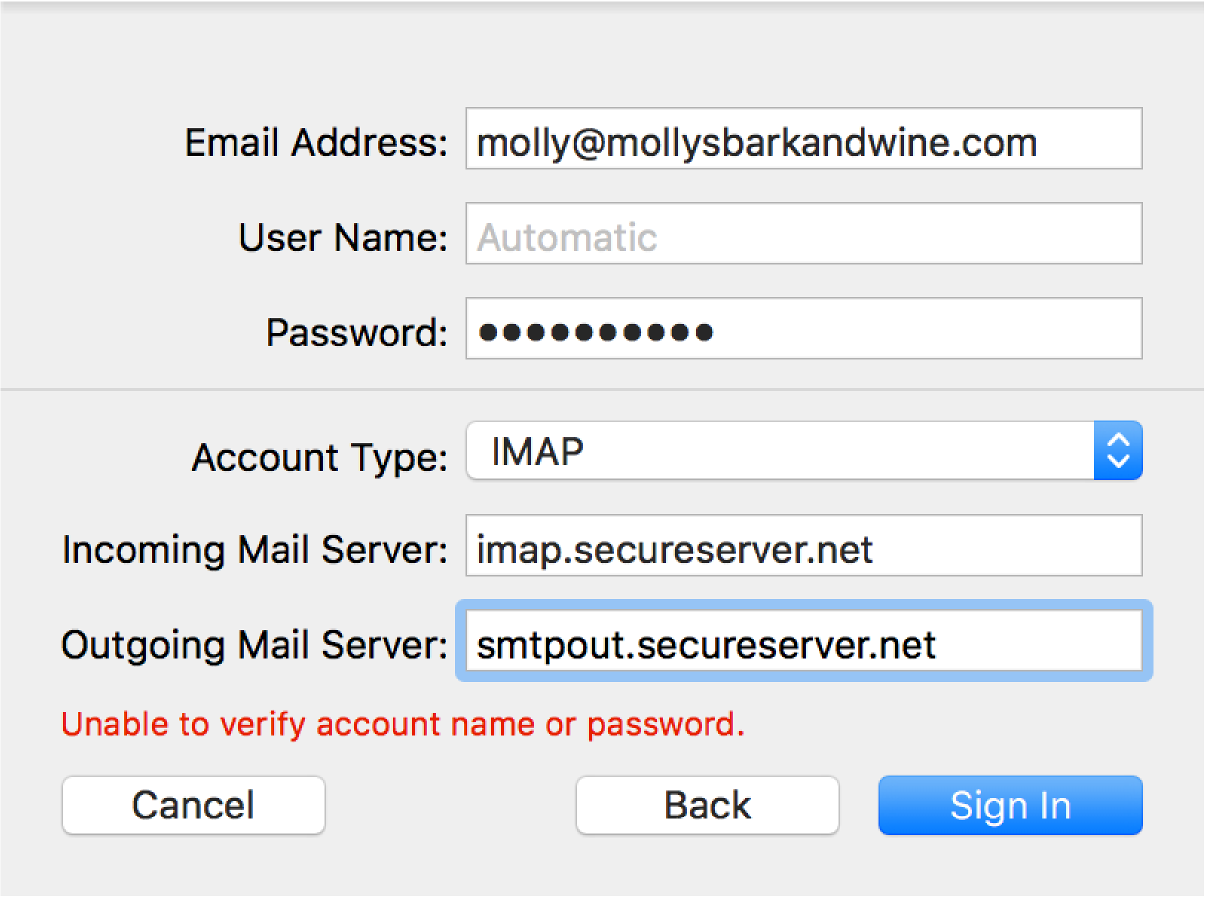
Amit Agarwal

blog: labnol.org work: ctrlq.org

appraisal.pdf (4K)

Step 4: You might want to include someone else in your email to ‘keep them in the loop’. You can do this by clicking **Cc** or **Bcc**, which will open another field. ‘Cc’ means ‘carbon copy’ and ‘Bcc’ means ‘blind carbon copy’. Adding an email address to the ‘Cc’ field means that that person will receive a copy of the email and all the other recipients will see their email address. If an email address is put into the 'Bcc’ field, the person will get a copy of the email but no other recipient will see that address.

If you are sending the same email to lots of different people, it’s a good idea to put all the email addresses in the ‘Bcc’ field to keep your ‘mailing list’ confidential. That way, there’s no chance that it could fall into the hands of a spammer or hacker.



Email Address: molly@mollysbarkandwine.com

User Name: Automatic

Password: ●●●●●●●●●●

Account Type: IMAP

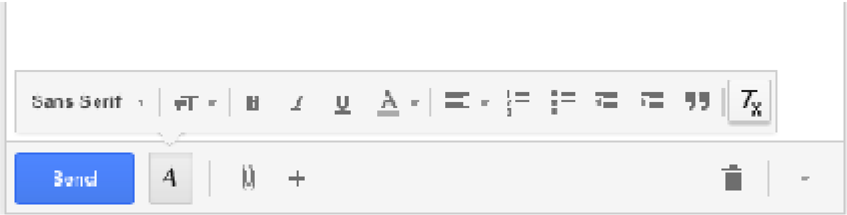
Incoming Mail Server: imap.secureserver.net

Outgoing Mail Server: smtpout.secureserver.net

Unable to verify account name or password.

Step 5: The subject field allows you to give the recipient an idea of the topic of your email, like a heading. You don't have to put anything in the subject box, but it can help when viewing and sorting email.

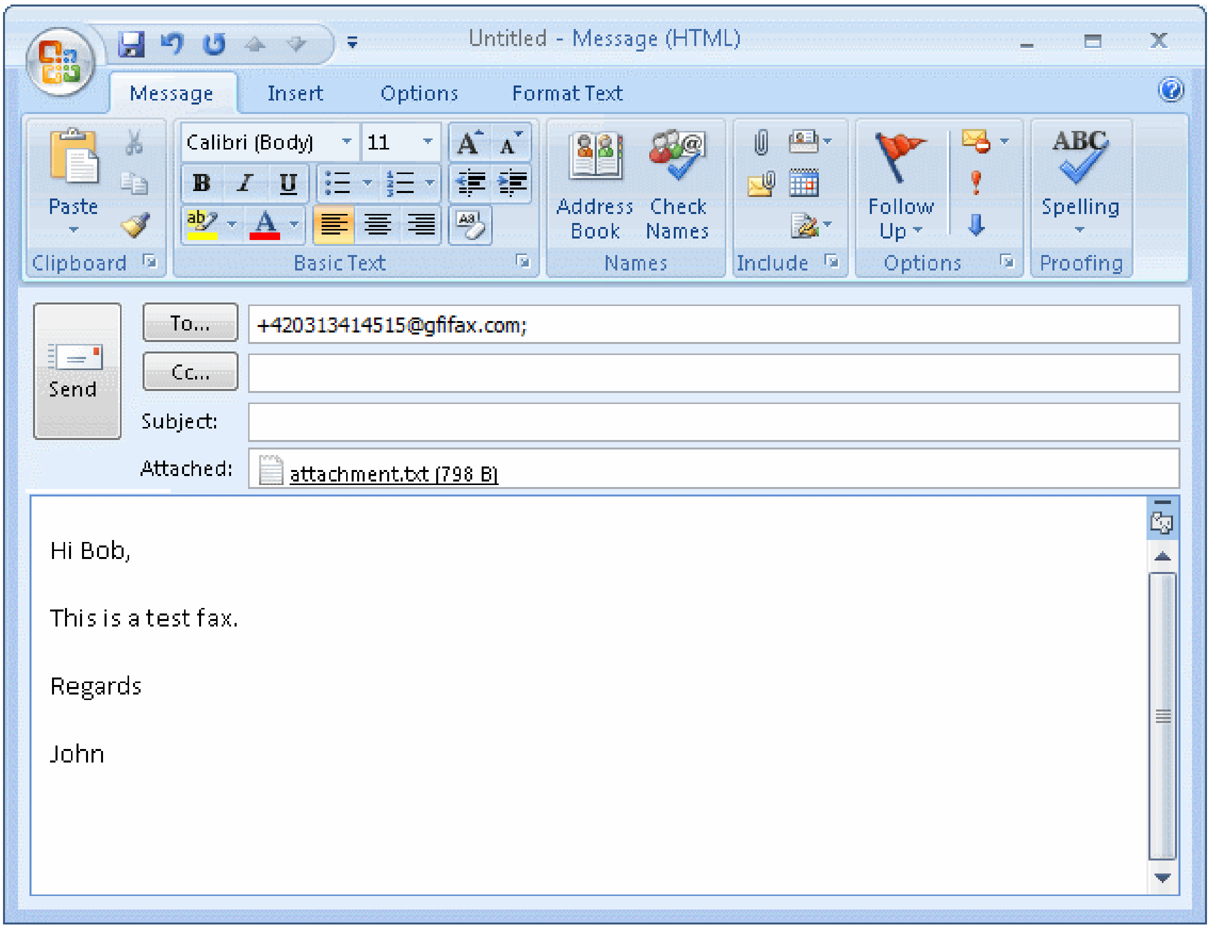
Step 6: Email text can be formatted in a similar way to text in a word document. You can change the font style, colour and size using the formatting icons. You can also create bullet points and check the spelling of your email. Choose your formatting from the menu shown.



Step 7: Type your message in the main body field of your email.

You can format your email using the options that are available on the toolbar. To add a link in the body of your email click on the insert link icon, then add the **'Text to display'** and then a web or email address, finally click **OK**.

Step 8: When you're happy with your email, click the blue **Send** button at the bottom of the compose window.



To... +420313414515@gfifax.com;

Attached: attachment.txt (798 B)

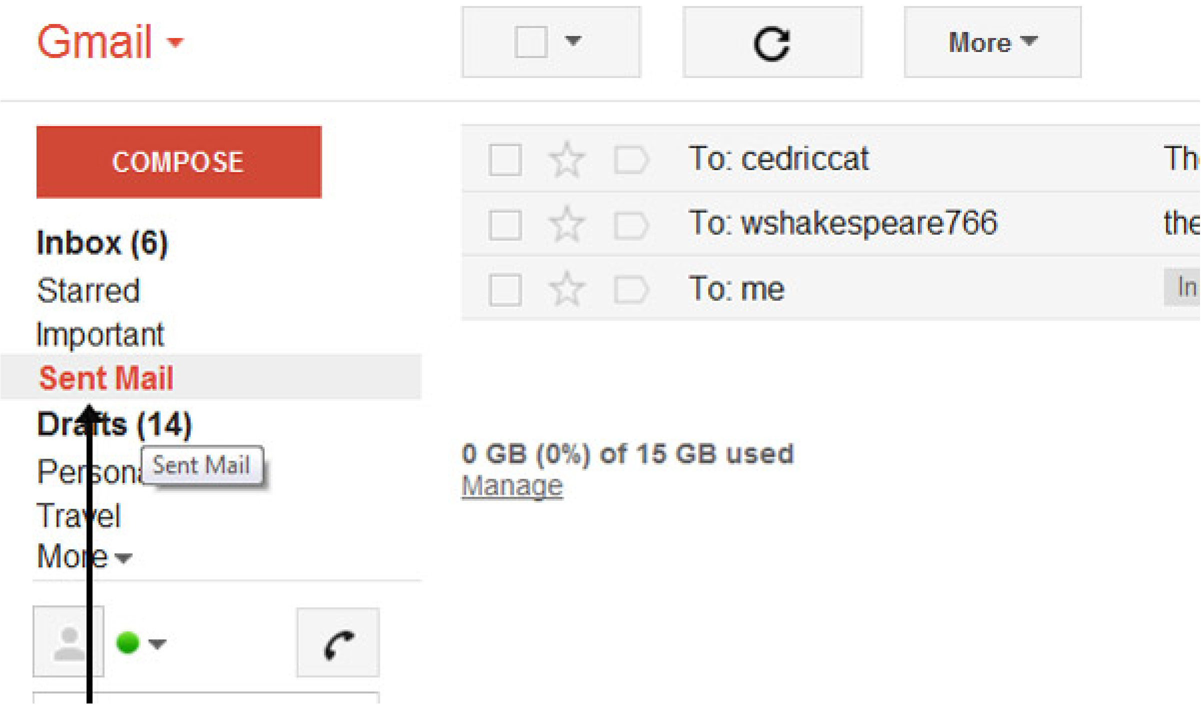
Hi Bob,

This is a test fax.

Regards

John

Step 9: The email you've sent will now be stored in the 'Sent Mail' folder on your Gmail dashboard. You may have to run your mouse pointer over the Inbox folder link to see the other folders.



Step 10: You may start an email but then decide to come back to it later rather then sending it straightaway. Gmail saves your drafts automatically. So you can simply close the email and the unfinished email will be saved to your 'Drafts' folder. When you decide that you're ready to send it, you can retrieve it from the 'Drafts' folder by clicking **Drafts** and then clicking the correct item in the 'Drafts' folder list. Finish the email and click **Send** as normal.

The email protocol is the process by which an email desktop/mobile client accesses the information (From and To) contained in an email account (your email address). Each of these protocols which are governed by universal standards (at the core level) have a specific set of capabilities in terms of how data is handled at the server level and stored at the local level.

E-mail Protocols:

What protocols are supported by a given email desktop/mobile client depends solely on the developers of the given client. It has nothing to do with what is or is not supported at the server level. For example, Thunderbird cannot support Outlook.com using the EAS protocol and it can't access a Microsoft Exchange account natively without an addin. Thunderbird could access an Exchange account via either the Imap or POP protocols but that is different then accessing the server using the native Exchange protocol.

By the same token, not every email service provider supports every emai protocol. With the exception of Outlook.com and hosted Exchange accounts, generally, only POP3 and IMAP are supported by a given provider.

In terms of contact/calendar data, neither the POP3 or IMAP protocol supports these data items. In Outlook (desktop), 3rd party addins are required to access such things as Google or iCloud based contact/calendar items. In the case of Google, this would required the Google App Sync addin (provided by Google with paid Google accounts). Apple provides an Outlook addin to connect to iCloud data. There are also 3rd party programs available that provide a wide-range of extended features such as Sync2 from 4Team or CompanionLink for Outlook from the CompanionLink folks.

The email clients on some devices are able to directly connect to contact/calendar info from various providers such as Google, Yahoo etc. This is done via extensions to the email client no different then installing an Outlook addin to accomplish the same thing when using Outlook (desktop).

What is POP3?

Post Office Protocol version 3 (POP3) is a standard mail protocol used to receive emails from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline.

POP is the most common Internet standard for receiving email. When a POP email client retrieves messages, the messages are downloaded from Comcast's secure mail server in the cloud onto your computer. The messages are normally deleted from the cloud, although you can configure your email client to not delete downloaded messages for a short period -- for example, a few days.

POP is best used when you are only using one computer or device to access your email. If you plan to also use webmail and/or receive email on a mobile device, POP may not be the best option for you. POP is only used for receiving email and is commonly used in combination with the SMTP email protocol to allow email clients to both send and receive messages.

POP is an e-mail only protocol. Items such as contacts, appointments or tasks cannot be managed or accessed using POP.

If you're using POP to receive Comcast Business email, Comcast's Business Security Assurance Team recommends configuring your email client's incoming server port number to 995 using an SSL-encrypted connection.

What is IMAP?

The Internet Message Access Protocol (IMAP) is a mail protocol used for accessing email on a remote web server from a local client. IMAP and POP3 are the two most commonly used Internet mail protocols for retrieving emails. Both protocols are supported by all modern email clients and web servers.

IMAP is another common Internet standard for retrieving email. When an IMAP email client retrieves messages, it downloads a copy of messages from the cloud to your computer but leaves a copy in the cloud as well. The email client periodically synchronizes your mailbox between the cloud and the email client. When a message is deleted from the mailbox on the email client, it will be deleted from the cloud the next time the two are synchronized.

By keeping a copy of each message on the server, IMAP facilitates access to your email from more than one computer or email client -- for example, Outlook on an office computer and via webmail from a home computer. IMAP is only used for receiving email and is commonly used in combination with the SMTP email protocol to provide both send and receive capabilities to email clients.

IMAP is an email only protocol. Items such as contacts, appointments and tasks cannot be managed or accessed using IMAP.

If you are using IMAP to receive Comcast Business email, Comcast's Business Security Assurance Team recommends configuring your email client's incoming server port number to 993 using an SSL encrypted connection.

What is SMTP?

Simple Mail Transfer Protocol (SMTP) is the standard protocol for sending emails across the Internet. SMTP uses TCP port 25 or 2525 and sometimes you can have problems sending your messages in case your ISP has closed port 25 (How to check if port 25 is open?). To determine the SMTP server for a given domain name, the MX (Mail eXchange) DNS record is used.

SMTP is the most common internet standard for sending email. When using SMTP, email is sent from your email client through Comcast's secure mail servers to the recipient. SMTP is only used for sending email and is commonly used in combination with either POP or IMAP to provide both send and receive capabilities to email clients.

Comcast's Business Security Assurance Team recommends configuring your email client's outgoing mail server to secured port 587. You must have a Comcast email address for authentication purposes if you are using the Comcast SMTP server to send email.

What is MAPI?

MAPI stands for Messaging Application Programming Interface. MAPI is a proprietary Microsoft protocol that allows the Microsoft Outlook email client to fully utilize all of the features of an Exchange server including email, shared address books, calendars and public folders. When Outlook is configured as a MAPI client, also known as an Exchange client, email is stored in the cloud on Comcast's secure mail server with a copy on your computer. Messages retained in the cloud are accesible via webmail from any internet connected computer.

With MAPI, you can move messages from the cloud into a local file on your computer called a .PST file, a process through which copies of messages are deleted from the cloud and stored on your computer. This can allow for valuable storage space and help you create backups of your business-critical emails.

Outlook 2010 allows up to 13 MAPI/Exchange email accounts at a time. As an alternative, you can instead use the POP or IMAP protocol to retreive email.

EAS – (Exchange ActiveSync)

The EAS protocol began life as the mobile protocol used to synchronize with Exchange servers and then being made available (for a fee) to mobile vendors to standardize the way to connect with Outlook.com (formerly Hotmail). EAS was then incorporated directly into Outlook '2013 to connect directly with an Outlook.com account eliminating the need for also having to install the Outlook Hotmail Connector previously required in earlier versions of Outlook.

Like MAPI, it supports synchroization of contacts, calendar and tasks but does not support sync'ing of things like notes and drafts. A major misunderstood sore-point is the lack of ability to sync Contact Groups created via the Outlook.com web interface to Outlook Contact Groups since the two employ radically different approaches. This is a limitation of the EAS protocol itself (at the time of writing). Outlook.com "Contact Groups" are created with the use of "categories" whereas Contact Groups created in Outlook (the desktop client) are created as special contact item types with a specific MessageClass (IPM.DistList) making it compatible with all earlier versions of Outlook using the MAPI interface via the Hotmail Outlook Connector. In short, you cannot synchornize Contact Groups using an EAS Outlook.com account and Outlook '2013.

The limitation related to Contact Groups (along with a number of other issues) disappears once the Outlook.com has been migrated. It should be noted that Outlook.com accounts originally configured using the Exchange ActiveSync protocol will continue to work normally after the account has been migrated on the server side. The account will need to be readded to the profile and Outlook will automatically use the <Exchange? protocol and re-sync all the data. The old Outlook.com can then be reomved. To determine if your Outlook.com account has been migrated, log ito the account using your browser and if it shows anything other than <Outlook.com> (i.e. Outlook Mail or Outlook EMail), the account has been moved to the new platform.

What is the main difference between POP3, IMAP & MAPI?

The POP3 protocol assumes that there is only one client connected to the mailbox. In contrast, the IMAP protocol allows simultaneous access by multiple clients. IMAP is suitable for you if your mailbox is about to be managed by multiple users.

POP3 is also a standard protocol but unlike IMAP4, it downloads the emails into the local computer. MAPI is a proprietary technology from Microsoft which uses RPC-based communication to communicate with a MAPI-based mail server like Exchange. IMAP4 is advance than POP3. An MAPI account provides almost the same functionality of a IMAP account, but it also have a live connection to the server.

MAPI is a messaging system that is used by Microsoft Outlook and Exchange for email, contacts, appointments, tasks, sticky notes, etc. It is based on a proprietary technology from

Microsoft which uses RPC-based communication to communicate with a MAPI-based mail server like Exchange.

It comprises of a standard set of C language functions that are stored in a program library which is known as a dynamic link library. It also provides the synchronization feature by which one can sync and view sub folders, custom folders, calendar, etc. on different devices and computers. It defines the following three services –

- Address book – a database which contains addressing information.
- Transport – assists in interaction between different devices.
- Message store – stores messages that consists folders and sub-folders.

In Short: Which Do I Use to Set Up My Email?

Depending on your personal style of communicating your email provider, you can pretty quickly narrow down how you should use your email.

- If you use check your email from a lot of devices, phones, or computers, use a webmail service or set up your email clients to use IMAP.
- If you use mostly webmail and want your phone or iPad to sync with your webmail, use IMAP, as well.
- If you're using one email client on one dedicated machine (say, in your office), you might be fine with POP3, but we'd still recommend IMAP.
- If you have a huge history of email and you're using an old mail provider without a lot of drive space, you may want to use POP3 to keep from running out of space on the remote email server.

Mailing List:

An **electronic mailing list** or **email list** is a special use of email that allows for widespread distribution of information to many Internet users. It is similar to a traditional mailing list – a list of names and addresses – as might be kept by an organization for sending publications to its members or customers, but typically refers to four things:

- a list of email addresses,
- the people ("subscribers") receiving mail at those addresses,
- the publications (email messages) sent to those addresses, and
- a *reflector*, which is a single email address that, when designated as the recipient of a message, will send a copy of that message to all of the subscribers.

File Transfer Protocol (FTP) :

is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.[1] FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[2][3] Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as web page editors.

The **File Transfer Protocol** (**FTP**) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.[1] FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[2][3] Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as web page editors

- The **File Transfer Protocol** (**FTP**) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.
- FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.[1] FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).
- The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[2][3] Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as web page editors.
- The **File Transfer Protocol** (**FTP**) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

- FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server.[1] FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).
- The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[2][3] Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as web page editors.

World Wide Web (WWW):-

World Wide Web:

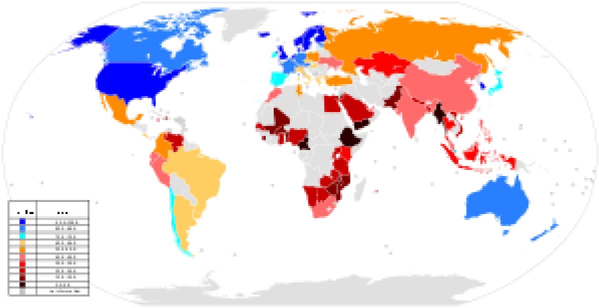
The **World Wide Web** (abbreviated **WWW** or **the Web**) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and accessible via the Internet.[1] English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser in 1990 while employed at CERN in Switzerland.[2][3] The browser was released outside CERN in 1991, first to other research institutions starting in January 1991 and to the general public on the Internet in August 1991.

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet.[4][5][6] Web pages are primarily text documents formatted and annotated with Hypertext Markup Language (HTML).[7] In addition to formatted text, web pages may contain images, video, audio, and software components that are rendered in the user's web browser as coherent pages of multimedia content .

Embedded hyperlinks permit users to navigate between web pages. Multiple web pages with a common theme, a common domain name, or both, make up a website. Website content can largely be provided by the publisher, or interactively where users contribute content or the content depends upon the users or their actions. Websites may be mostly informative, primarily for entertainment, or largely for commercial, governmental, or non-governmental organisational purposes.

World Wide Web (later renamed to **Nexus** to avoid confusion between the software and the World Wide Web) was the first web browser[1] and editor.[2] It was discontinued in 1994. At the time it was written, it was the sole web browser in existence,[1] as well as the first WYSIWYG HTML editor.

The source code was released into the public domain on April 30, 1993.[3][4] Some of the code still resides on Tim Berners-Lee's NeXT Computer in the CERN museum and has not been recovered due to the computer's status as a historical artifact.[*citation needed*] To coincide with the 20th anniversary of the research centre giving the web to the world, a project began in dexter at CERN to preserve this original hardware and software associated with the birth of the Web.



World Wide Web

Berners-Lee wrote what would become known as WorldWideWeb on a NeXT Computer during the second half of 1990, while working for CERN. The first successful build was completed by December 25, 1990, after only two months of development.Successive builds circulated among Berners-Lee's colleagues at CERN before being released to the public, by way of Internet newsgroups, in August 1991. By this time, several others, including Bernd Pollermann, Robert Cailliau, Jean-François Groff, and visiting undergraduate student Nicola Pellow – who wrote the Line Mode Browser – were involved in the project.

Berners-Lee proposed different names for his new application: *The Mine of Information* and *The Information Mesh*were proposals. At the end *WorldWideWeb* was chosen, but later renamed to *Nexus* to avoid confusion between the World Wide Web and the web browser.

The team created so called "passive browsers" which do not have the ability to edit because it was hard to port this feature from the NeXT system to other operating systems. Porting to the X Window System (X) was not possible as nobody on the team had experience with X.[

Berners-Lee and Groff later adapted many of WorldWideWeb's components into a C programming language version, creating the libwww API.

A number of early browsers appeared, notably ViolaWWW. They were all eclipsed by Mosaic in terms of popularity, which by 1993 had replaced the WorldWideWeb program. Those involved in its creation had moved on to other tasks, such as defining standards and guidelines for the further development of the World Wide Web (e.g. HTML, various communication protocols).

On April 30, 1993, the CERN directorate released the source code of WorldWideWeb into the public domain. Several versions of the software are still available on the web.[10] Berners-Lee initially considered releasing it under the GNU General Public License, but eventually opted for public domain to maximize corporate support.

Features:

Since WorldWideWeb was developed on and for the NeXTSTEP platform, the program uses many of NeXTSTEP's components – WorldWideWeb's layout engine was built around NeXTSTEP's Text class.[1]

WorldWideWeb is capable of displaying basic style sheets,[4] downloading and opening any file type supported by the NeXT system (PostScript,[2][4] movies, and sounds[4]), browsing newsgroups, and spellchecking. In earlier versions, images are displayed in separate windows, until NeXTSTEP's Text class gained support for Image objects.[4] WorldWideWeb is able to use different protocols: FTP, HTTP, NNTP, and local files. Later versions are able to display inline images.[1]

The browser is also a WYSIWYG editor. It allows the simultaneous editing and linking of many pages in different windows. The functions "Mark Selection", which creates an anchor, and "Link to Marked", which makes the selected text an anchor linking to the last marked anchor, allow the creation of links. Editing pages remotely is not possible, as the HTTP PUT method had not yet been implemented during the period of the application's active development. Files can be edited in a local file system which is in turn served onto the Web by an HTTP server.

WorldWideWeb's navigation panel contain Next and Previous buttons that automatically navigate to the next or previous link on the last page visited, similar to Opera's Rewind and Fast Forward buttons; i.e., if one navigated to a page from a table of links, the Previous button would cause the browser to load the previous page linked in the table.] This is useful for web pages which contain lists of links. Many still do, but the user interface link-chaining was not adopted by other contemporary browser writers, and it only gained popularity later. An equivalent functionality is nowadays provided by connecting web pages with explicit navigation buttons repeated on each webpage among those links, or with typed links in the headers of the page. This places more of a burden on web site designers and developers, but allows them to control the presentation of the navigation links.

WorldWideWeb does not have features like bookmarks, but a similar feature was presented in the browser: if a link should be saved for later use linking it to the user's own home page (start page), the link is remembered in the same fashion as a bookmark. The ability to create more home pages was implemented, similar to folders in the actual web browsers bookmarks.

Concept of Search Engines:-

A of being searched by a web search engine is generally described as the deep web. **web search engine** is a software system that is designed to search for information on the World Wide Web. The search results are generally presented in a line of results often referred to as search engine results pages(SERPs). The information may be a mix of web pages, images, and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler. Internet content that is not capable of being searched by a web search engine is generally described as the deep web.

History:-

Internet search engines themselves predate the debut of the Web in December 1990. The Who is user search dates back to 1982 [1] and the Knowbot Information Service multi-network user search was first implemented in 1989.[2] The first well documented search engine that searched content files, namely FTP files was Archie, which debuted on 10 September 1990.[3]

Prior to September 1993 the World Wide Web was entirely indexed by hand. There was a list of webservers edited by Tim Berners-Lee and hosted on the CERN webserver. One Google.nl snapshot of the list in 1992 remains,[4] but as more and more web servers went online the central list could no longer keep up. On the NCSA site, new servers were announced under the title "What's New!"[5]

The first tool used for searching content (as opposed to users) on the Internet was Archie.[6] The name stands for "archive" without the "v".

It was created by Alan Emtage, Bill Heelan and J. Peter Deutsch, computer science students at McGill University in Montreal, Quebec, Canada. The program downloaded the directory listings of all the files located on public anonymous FTP (File Transfer Protocol) sites, creating a searchable database of file names; however, Archie Search Engine did not index the contents of these sites since the amount of data was so limited it could be readily searched manually.

The rise of Gopher (created in 1991 by Mark McCahill at the University of Minnesota) led to two new search programs, Veronica and Jughead. Like Archie, they searched the file names and titles stored in Gopher index systems. Veronica (*V*ery *E*asy *R*odent-*O*riented *N*et-wide *I*ndex to *C*omputerized *A*rchives) provided a keyword search of most Gopher menu titles in the entire Gopher listings. Jughead (*J*onzy's *U*niversal *G*opher *H*ierarchy *E*xcavation *A*nd *D*isplay) was a tool for obtaining menu information from specific Gopher servers. While the name of the search engine "Archie Search Engine" was not a reference to the Archie comic book series, "Veronica" and "Jughead" are characters in the series, thus referencing their predecessor.

In the summer of 1993, no search engine existed for the web, though numerous specialized catalogues were maintained by hand. Oscar Nierstrasz at the University of Geneva wrote a series of Perl scripts that periodically mirrored these pages and rewrote them into a standard format. This formed the basis for W3Catalog, the web's first primitive search engine, released on September 2, 1993.[7]

In June 1993, Matthew Gray, then at MIT, produced what was probably the first web robot, the Perl-based World Wide Web Wanderer, and used it to generate an index called 'Wandex'. The purpose of the Wanderer was to measure the size of the World Wide Web, which it did until late 1995. The web's second search engine Aliweb appeared in November 1993. Aliweb did not use a web robot, but instead depended on being notified by website administrators of the existence at each site of an index file in a particular format.

NCSA's Mosaic™ - Mosaic (web browser) wasn't the first Web browser. But it was the first to make a major splash. In November 1993, Mosaic v 1.0 broke away from the small pack of existing browsers by including features—like icons, bookmarks, a more attractive interface, and pictures—that made the software easy to use and appealing to "non-geeks."

JumpStation (created in December 1993[8] by Jonathon Fletcher) used a web robot to find web pages and to build its index, and used a web form as the interface to its query program. It was thus the first WWW resource-discovery tool to combine the three essential features of a web search engine (crawling, indexing, and searching) as described below. Because of the limited resources available on the platform it ran on, its indexing and hence searching were limited to the titles and headings found in the web pages the crawler encountered.

One of the first "all text" crawler-based search engines was WebCrawler, which came out in 1994. Unlike its predecessors, it allowed users to search for any word in any webpage, which has become the standard for all major search engines since. It was also the first one widely known by the public. Also in 1994, Lycos(which started at Carnegie Mellon University) was launched and became a major commercial endeavor.

Soon after, many search engines appeared and vied for popularity. These included Magellan, Excite, Infoseek, Inktomi, Northern Light, and AltaVista. Yahoo! was among the most popular ways for people to find web pages of interest, but its search function operated on its web directory, rather than its full-text copies of web pages. Information seekers could also browse the directory instead of doing a keyword-based search.

In 1996, Netscape was looking to give a single search engine an exclusive deal as the featured search engine on Netscape's web browser. There was so much interest that instead Netscape struck deals with five of the major search engines: for $5 million a year, each search engine would be in rotation on the Netscape search engine page. The five engines were Yahoo!, Magellan, Lycos, Infoseek, and Excite.[9][10]

Google adopted the idea of selling search terms in 1998, from a small search engine company named goto.com. This move had a significant effect on the SE business, which went from struggling to one of the most profitable businesses in the internet.[11]

Search engines were also known as some of the brightest stars in the Internet investing frenzy that occurred in the late 1990s.[12] Several companies entered the market spectacularly, receiving record gains during their initial public offerings. Some have taken down their public search engine, and are marketing enterprise-only editions, such as Northern Light. Many search engine companies were caught up in the dot-com bubble, a speculation-driven market boom that peaked in 1999 and ended in 2001.

- Around 2000, Google's search engine rose to prominence.[13] The company achieved better results for many searches with an innovation called PageRank, as was explained in the paper *Anatomy of a Search Engine* written by Sergey Brin and Larry Page, the later founders of Google.[14] This iterative algorithm ranks web pages based on the number and PageRank of other web sites and pages that link there, on the premise that good or desirable pages are linked to more than others. Google also maintained a minimalist interface to its search engine. In contrast, many of its competitors embedded a search engine in a web portal. In fact, Google search engine became so popular that spoof engines emerged such as Mystery Seeker.

By 2000, Yahoo! was providing search services based on Inktomi's search engine. Yahoo! acquired Inktomi in 2002, and Overture (which owned AlltheWeb and AltaVista) in 2003. Yahoo! switched to Google's search engine until 2004, when it launched its own search engine based on the combined technologies of its acquisitions.

Microsoft first launched MSN Search in the fall of 1998 using search results from Inktomi. In early 1999 the site began to display listings from Looksmart, blended with results from Inktomi. For a short time in 1999, MSN Search used results from AltaVista instead. In 2004, Microsoft began a transition to its own search technology, powered by its own web crawler (called msnbot).

Microsoft's rebranded search engine, Bing, was launched on June 1, 2009. On July 29, 2009, Yahoo! and Microsoft finalized a deal in which Yahoo! Search would be powered by Microsoft Bing technology.

Searching the Web:

Introduction :

The plentiful content of the World-Wide Web is useful to millions. Some simply browse the Web through entry points such as Yahoo!. But many information seekers use a search engine to begin their Web activity. In this case, users submit a query, typically a list of keywords, and receive a list of Web pages that may be relevant, typically pages that contain the keywords.

In this paper we discuss the challenges in building good search engines, and describe some of the techniques that are useful. Many of the search engines use well-known information retrieval (IR) algorithms and techniques [55, 28]. However, IR algorithms were developed for relatively small and coherent collections such as newspaper articles or book catalogs in a (physical) library. The Web, on the other hand, is massive, much less coherent, changes more rapidly, and is spread over geographically distributed computers.

This requires new techniques, or extensions to the old ones, to deal with the gathering of the information, to make index structures scalable and efficiently updateable, and to improve the discriminating ability of search engines. For the last item, discriminating ability, it is possible to exploit the linkage among Web pages to better identify the truly relevant pages. There is no question that the Web is huge and challenging to deal with. Several studies have estimated the size of the Web and while they report slightly different numbers, most of 1 them agree that over a billion pages are available. Given that the average size of a Web page is around 5–10K bytes, just the textual data amounts to at least tens of terabytes. The growth rate of the Web is even more dramatic.

According to [41, 42], the size of the Web has doubled in less than two years, and this growth rate is projected to continue for the next two years. Aside from these newly created pages, the existing pages are continuously updated . For example, in our own study of over half a million pages over 4 months we found that about 23% of pages changed daily. In the .com domain 40% of the pages changed daily, and the half-life of pages is about 10 days (in 10 days half of the pages are gone, i.e., their URLs are no longer valid). In , we also report that a Poisson process is a good model for Web page changes. Later in Section 2, we will show how some of these results can be used to improve search engine quality. In addition to size and rapid change, the interlinked nature of the Web sets it apart from many other collections. Several studies aim to understand how the Web's linkage is structured and how that structure can be modeled . One recent study, for example, suggests that the link structure of the Web is somewhat like a "bow-tie" [11]. That is, about 28% of the pages constitute a strongly connected core (the center of the bow tie). About 22% form one of the tie's loops: these are pages that can be reached from the core but not vice versa. The other loop consists of 22% of the pages that can reach the core, but cannot be reached from it.

Crawling Web pages :-

The crawler module retrieves pages from the Web for later analysis by the indexing module. As discussed in the introduction, a crawler module typically starts off with an initial set of URLs S0. Roughly, it first places S0 in a queue, where all URLs to be retrieved are kept and prioritized. From this queue, the crawler gets a URL (in some order), downloads the page, extracts any URLs in the downloaded page, and puts the new URLs in the queue. This process is repeated until the crawler decides to stop. Given the enormous size and the change rate of the Web, many issues arise, including the following:

1. **What pages should the crawler download?**

.In most cases, the crawler cannot download all pages on the Web. Even the most comprehensive search engine currently indexes a small fraction of the entire Web . Given this fact, it is important for the crawler to carefully select the pages and to visit "important" pages first by prioritizing the URLs in the queue properly, so that the fraction of the Web that is visited (and kept up-to-date) is more meaningful.

2. **How should the crawler refresh pages?**

Once the crawler has downloaded a significant number of pages, it has to start revisiting the downloaded pages in order to detect changes and refresh the downloaded collection. Because Web pages are changing at very different rates the crawler needs to carefully decide what page to revisit and what page to skip, because this decision may significantly impact the "freshness" of the downloaded collection. For example, if a certain page rarely changes, the crawler may want to revisit the page less often, in order to visit more frequently changing ones.

3. **How should the load on the visited Web sites be minimized?**

   When the crawler collects pages from the Web, it consumes resources belonging to other organizations.

For example, when the crawler downloads page p on site S, the site needs to retrieve page p from its file system, consuming disk and CPU resource. Also, after this retrieval the page needs to be transferred through the network, which is another resource shared by multiple organizations. The crawler should minimize its impact on these resources . Otherwise, the administrators of the Web site or a particular network may complain and sometimes completely block access by the crawler.

4. **How should the crawling process be parallelized?**
   Due to the enormous size of the Web, crawlers often run on multiple machines and download pages in parallel . This parallelization is often necessary in order to download a large number of pages in a reasonable amount of time. Clearly these parallel crawlers should be coordinated properly, so that different crawlers do not visit the same Web site multiple times, and the adopted crawling policy should be strictly enforced. The coordination can

incur significant communication overhead, limiting the number of simultaneous crawlers.

HTTP (Hypertext Transfer Protocol):

The **Hypertext Transfer Protocol** (**HTTP**) is an application protocol for distributed, collaborative, and hyper media information systems. HTTP is the foundation of data communication for the World Wide Web.

Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

Development of HTTP was initiated by Tim Berners-Lee at CERN in 1989. Standards development of HTTP was coordinated by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C), culminating in the publication of a series of Requests for Comments (RFCs). The first definition of HTTP/1.1, the version of HTTP in common use, occurred in RFC 2068 in 1997, although this was made obsolete by RFC 2616 in 1999 and then again by the RFC 7230 family of RFCs in 2014.

A later version, the successor HTTP/2, was standardized in 2015, and is now supported by major web servers and browsers over TLS using ALPN extension where TLS 1.2 or newer is required.

HTTP Session :

An HTTP session is a sequence of network request-response transactions. An HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a server (typically port 80, occasionally port 8080; see List of TCP and UDP port numbers). An HTTP server listening on that port waits for a client's request message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own. The body of this message is typically the requested resource, although an error message or other information may also be returned.

HISTORY:-

HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the *client* and an application running on a computer hosting a website may be the *server*. The client submits an HTTP *request*message to the server. The server, which provides *resources* such as HTML files and other content, or performs other functions on behalf of the client, returns a *response* message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

A web browser is an example of a *user agent* (UA). Other types of user agent include the indexing software used by search providers (web crawlers), voice browsers, mobile apps, and other software that accesses, consumes, or displays web content.

HTTP is designed to permit intermediate network elements to improve or enable communications between clients and servers. High-traffic websites often benefit from web cache servers that deliver content on behalf of upstream servers to improve response time. Web browsers cache previously accessed web resources and reuse them when possible to reduce network traffic. HTTP proxy servers at private network boundaries can facilitate communication for clients without a globally routable address, by relaying messages with external servers.

HTTP is an application layer protocol designed within the framework of the Internet protocol suite. Its definition presumes an underlying and reliable transport layerprotocol,[4] and Transmission Control Protocol (TCP) is commonly used. However HTTP can be adapted to use unreliable protocols such as the User Datagram Protocol (UDP), for example in HTTPU and Simple Service Discovery Protocol (SSDP).

HTTP resources are identified and located on the network by Uniform Resource Locators (URLs), using the Uniform Resource Identifiers (URI's) schemes *http* and *https*. URIs and hyperlinks in HTML documents form inter-linked hypertext documents.

HTTP/1.1 is a revision of the original HTTP (HTTP/1.0). In HTTP/1.0 a separate connection to the same server is made for every resource request. HTTP/1.1 can reuse a connection multiple times to download images, scripts, stylesheets, *etc* after the page has been delivered. HTTP/1.1 communications therefore experience less latency as the establishment of TCP connections presents considerable overhead.

Web Protocols:-

Below is a list of protocols used for the world wide web:

- **ARP:** Address Resolution Protocol
- **DHCP:** Dynamic Host Configuration Protocol
- **DNS:** Domain Name Service
- **DSN:** Data Source Name
- **FTP:** File Transfer Protocol
- **HTTP:** Hypertext Transfer Protocol
- **IMAP:** Internet Message Access Protocol
- **ICMP:** Internet Control Message Protocol
- **IDRP:** ICMP Router-Discovery Protocol
- **IP:** Internet Protocol
- **IRC:** Internet Relay Chat Protocol
- **POP3:** Post Office Protocol version 3

- **PAR:** Positive Acknowledgment and Retransmission
- **RLOGIN:** Remote Login
- **SMTP:** Simple Mail Transfer Protocol
- **SSL:** Secure Sockets Layer
- **SSH:** Secure Shell
- **TCP:** Transmission Control Protocol
- **TELNET:** TCP/IP Terminal Emulation Protocol
- **UPD:** User Datagram Protocol
- **UPS:** Uninterruptible Power Supply

Web Publishing:-

Web publishing is the process of publishing original content on the Internet. The process includes building and uploading websites, updating the associated web pages, and posting content to these web pages online. Web publishing comprises of personal, business, and community websites in addition to e-books and blogs.

The content meant for web publishing can include text, videos, digital images, artwork, and other forms of media.

Publishers must possess a web server, a web publishing software, and an Internet connection to carry out web publishing. Web publishing is also known as online publishing.

A publisher requires three things to publish content on the Internet:

Website development software
Internet connection
A web server to host the website

The website development software can be a professional web design application like Dreamweaver or a straightforward web-based content management system like WordPress. Publishers require an Internet connection to upload the content to the web server.

Major sites may utilize a dedicated server to host them; however, many smaller websites usually reside on shared servers that host an array of websites.

Since web publishing does not demand physical materials like ink and paper, it costs practically nothing to publish the content.

Thus, anyone fulfilling the above three requirements can become a web publisher. In addition, web publishing brings countless visitors as the published contents are accessed by global visitors. These benefits of web publishing opened a new era of personal publishing, which was unimaginable before.

E-book and blog publishers utilize almost the same web publishing tools used by the website developers. People who do not have the required web publishing skills seek the services of professional web publishing individuals or organizations to host, maintain, and modify their websites, e-books and blogs.

Posting updates on social media sites like Twitter, Facebook, etc. is usually not considered web publishing; instead, web publishing usually refers to uploading original content to unique websites.

Domain name Registration:-

Domain registration is the process of registering a domain name, which identifies one or more IP addresses with a name that is easier to remember and use in URLs to identify particular Web pages. The person or business that registers domain name is called the domain name registrant.

Requirements for Domain Registration:

Domain registration requires utilizing the services of a domain name registrar, an ICANN or national ccTLD accredited company that has the authority to register domain names. Registrars help individuals and organizations register a domain name that has an extension like .com, .org, .net, .info, .biz, .us, .mobi, .name, .pro, .tv, etc.

Space On Host Server For Web Site:-

The web space, also known as storage space or disk space, generally refers to the amount of space on a web server that is allocated to website owners by the web hosting companies. It is made up of the total quantity of all text files, images, scripts, databases, emails and other files related to your website.

Web Space:-

There are several essential things a web hosting provider must actually provide. The first thing is a stable server, on which the websites should run. Once he has the physical machine, it's time to install suitable software, such as Mail SMTP server and DNS server software. And when everything is ready to go, the hosting provider must face the hardest task - to define his offers. And one of the features every client first looks for is the web space.

The web space, also known as storage space or disk space, generally refers to the amount of space on a web server that is allocated to website owners by the web hosting companies. It is made up of the total quantity of all text files, images, scripts, databases, emails and other files related to your website.

Having an idea of the web space demands for your web presence will help you choose the right web hosting plan configuration. Thus, you will feel secure about the online availability of your uploaded content and hence - for your web image. This will make your uploaded content always available online and your virtual profile - invulnerable to shortage in resources.

Web Space Functions:-

The web space can serve two basic purposes. In the first place, it allows you to upload file information (HTML files, image files, etc.) on the World Wide Web where it will be available at a global scale. Second, this resource enables you to store various files that are not visible to website visitors but play an important role for the proper functioning of your website.

Some of the popular 'invisible' files taking up web space on the server where your website is located are PHP files, database files and CGI program files. PHP files are stored on the server with a .php extension and are used for various important on-site activities such as order form processing for online stores, poll results management, etc. Databases, in turn, store data such as product codes, customer details, etc., which is retrieved by PHP scrips and CGI programs. CGI programs serve for processing data inputs from online forms, which require that the collected information be stored on the website's server.

Other web space occupying files worth mentioning include externally linked CSS files and JavaScript files. External CSS files, responsible for defining the style elements of a web page, are stored on the web hosting server and linked to each web page requiring them. JavaScript files, also linked to web pages needing them, lie in the basis of dynamic drop-down menus, visitor counters, etc., i.e. they work for increasing the interactivity of a website.

Log files are other website-related data containers that eat up your allocated web space. They contain important details about your website visitors' behavior, generated through requests sent from your website to the web server. The email accounts and the separate email messages are also considered as disk space consumers. The storage amount they occupy is usually added to the web space quota of the regular web hosting plans.

As you can see, the web space you need to launch a website on the World Wide Web is not used only for accommodating files visible to the visitor's eye. It also refers to storing files and programs that lie in the background of a website, but are responsible for supporting its interactivity, dynamic content, statistics and e-mail communication. In this light, when estimating your web space demands, you need to first total the quantity of your website files on your computer and then add approximately estimated space for databases, emails, log files, etc. In view of your website's future smooth expansion, you should add at least half of the whole estimated space to the calculation you have made so far. Now it will be much easier for you to find the right web hosting account configuration.

Web space measurement:-

The web space is generally measured in bytes, kilobytes (1,000 bytes), megabytes (1,000 kilobytes) and gigabytes (1,000 megabytes) on both personal computers and web servers. Since disk space has lately become a comparatively cheap web hosting resource, it is usually offered in gigabyte quantities with standard plans. Megabytes are represented with "MB" and gigabytes with "GB". A popular web hosting trick is to represent disk space in smaller units, in order to impress the customers.

Web Space with NTC Hosting:-

All NTC's web hosting offers come with enough disk space for any type of sites - from a small personal site to a complex company web page or a popular online e-commerce store. Disk space can be easily monitored at all times from both the Accounts usage table in the Web Hosting Control Paneland via a neat graph in the File Manager.

HTML:- Hypertext Markup Language

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.[4]

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as **<img />** and **<input />** directly introduce content into the page. Other tags such as **<p>** surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

In computer text processing, a markup language is a system for annotating a document in a way that is syntactically distinguishable from the text.[1] The idea and terminology evolved from the "marking up" of paper manuscripts, i.e., the revision instructions by editors, traditionally written with a blue pencil on authors' manuscripts.[*citation needed*] In digital media, this "blue pencil instruction text" was replaced by tags, that is, instructions are expressed directly by tags or "instruction text encapsulated by tags."

Examples include typesetting instructions such as those found in troff, TeX and LaTeX, or structural markers such as XML tags. Markup instructs the software that displays the text to carry out appropriate actions, but is omitted from the version of the text that users see.

Some markup languages, such as the widely used HTML, have pre-defined presentation semantics—meaning that their specification prescribes how to present the structured data. Others, such as XML, do not have them and are general purpose.

HyperText Markup Language (HTML), one of the document formats of the World Wide Web, is an instance of Standard Generalized Markup Language or SGML, and follows many of the markup conventions used in the publishing industry in the communication of printed work between authors, editors, and printers.

A **web page** (also written as **webpage**) is a document that is suitable for the World Wide Web and web browsers. A web browser displays a web page on a monitor or mobile device.

The web page usually means what is visible, but the term may also refer to a computer file, usually written in HTML or a comparable markup language. Web browsers coordinate various web resource elements for the written web page, such as style sheets, scripts, and images, to present the web page. Typical web pages provide hypertext that includes a navigation baror a sidebar menu linking to *other* web pages via hyperlinks, often referred to as *links*.

On a network, a web browser can retrieve a web page from a remote web server. The web server may restrict access to a private network such as a corporate intranet. The web browser uses the Hypertext Transfer Protocol (HTTP) to make such requests.

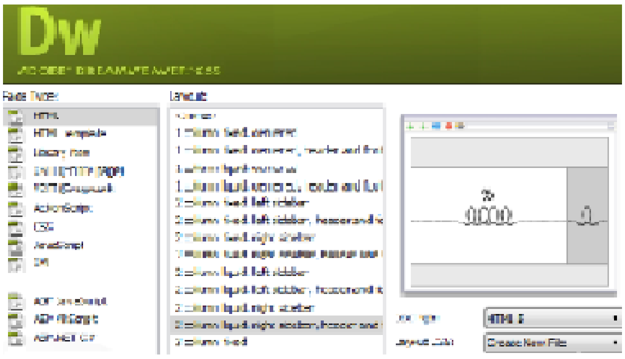
A *static* web page is delivered exactly as stored, as web content in the web server's file system. In contrast, a *dynamic* web page is generated by a web application, usually driven by server-side software. Dynamic web pages help the browser (the client) to enhance the web page through user input to the server.

In computing, a **web application** or **web app** is a client–server computer program which the client (including the user interface and client-side logic) runs in a web browser.[1] Common web applications include webmail, online retail sales, online auctions, wikis, instant messaging services and many other functions.

These useful HTML5 tools will aid various areas of web design and development:

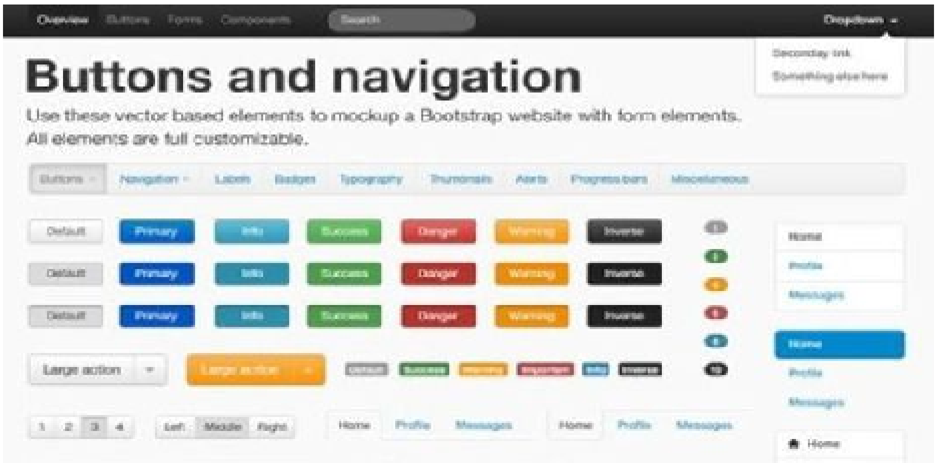
HTML5 seems to have been around forever but it was in fact only finalised in October 2014 – although it has been implemented for around five years prior. More and more of the web's legacy sites are switching to this markup language due to its functionality and ability to create responsive web design, and HTML5's involvement in new sites is virtually taken for granted – have a look at these great HTML5 examples.

However, HTML5 implementation is not as straightforward as it may seem, especially for new and inexperienced web designers. Neither skilled nor amateur web developers or designers can operate effectively without good HTML5 tools. Here we've provided a list of the most essential and best HTML5 tools to help create amazing modern websites.



Dreamweaver is still a great HTML5 tool

This is the most used HTML authoring tool for responsive website design. It combines a visual design surface and a code editor. Together with free HTML5 Pack extension, this tool provides more options, allowing web developers to easily create, deliver and optimize HTML5 content for diverse desktop and mobile platforms. Dreamweaver is available as a part of Adobe Creative Cloud2015, with plans starting from US$19.99 per month.

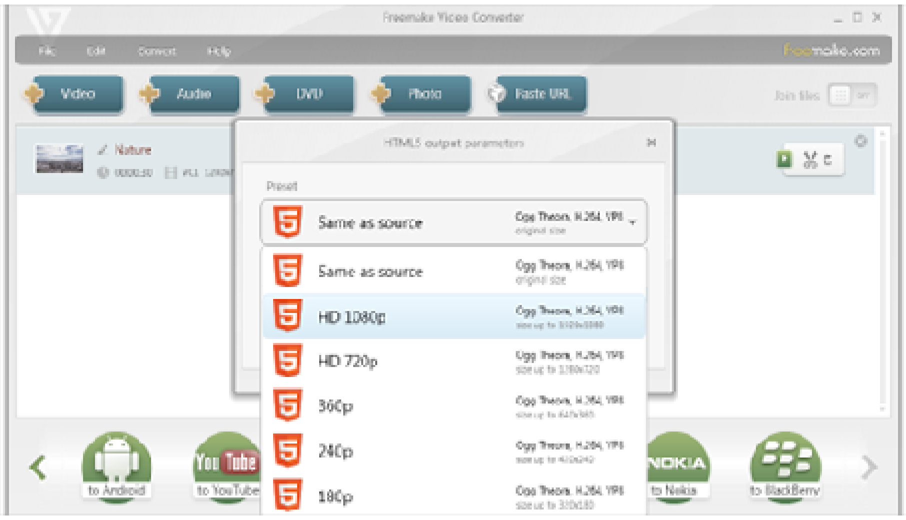


Buttons and navigation

Use these vector based elements to mockup a Bootstrap website with form elements.
All elements are full customizable.

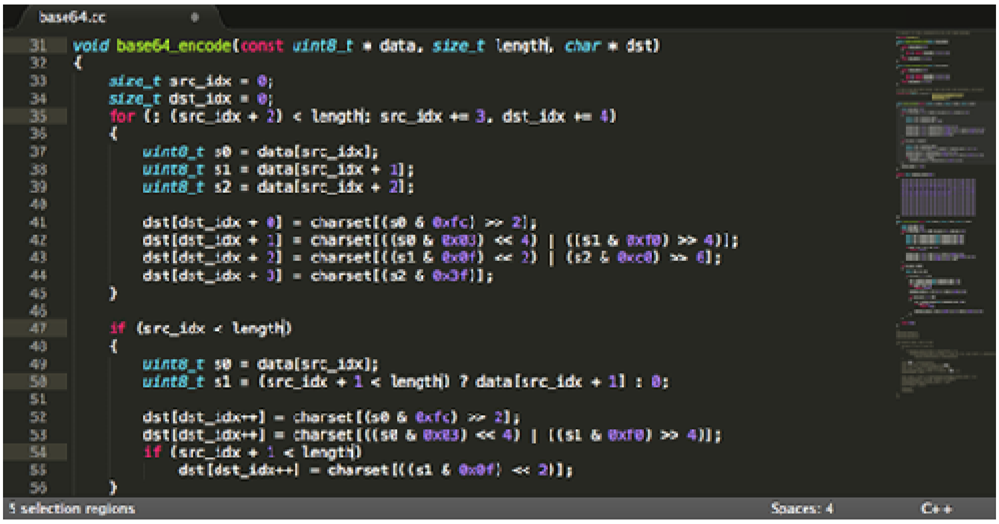
Bootstrap is free and open-source

Bootstrap is a free and open-source tool kit and framework for creating websites and web applications. It offers HTML and CSS-based design templates which you can tailor to your needs. Another alternative is HTML5 Boilerplate, which is along similar lines.
The tool provides web designers with a great deal typography elements, forms, buttons, navigation, as well as optional JavaScript extensions. To switch to HTML5 doctype, you just need to include it at the beginning of all your projects.



Create HTML5 video out of virtually any format

This free video converter will help web developers create HTML5-ready video. It may turn any desktop video or DVD to three HTML5 compatible formats: MP4/H.264, Theora/OGG, VP8/WebM. All you will need is to upload the resulted videos to your server and copy-paste the embed code the tool provides onto your webpage. That is an ideal solution for websites owners who prefer in-house video hosting to public video services like YouTube or Vimeo.



```
void base64_encode(const uint8_t * data, size_t length, char * dst)
{
    size_t src_idx = 0;
    size_t dst_idx = 0;
    for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = data[src_idx + 1];
        uint8_t s2 = data[src_idx + 2];

        dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
        dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
        dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
        dst[dst_idx + 3] = charset[(s2 & 0x3f)];
    }

    if (src_idx < length)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;

        dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
        dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
        if (src_idx + 1 < length)
            dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
    }
```

Sublime Text is a great multi-purpose text editor

This is a simple yet powerful advanced text editor for code and markup. But its simplicity is only on the surface. If you dig in just a bit, and you'll be able to find built-in plugins, clever auto-completion tricks, and more. To add HTML5 syntax and snippets to it, you need to install HTML5 bundle. Sublime Text may be downloaded and evaluated for free, however a $70 license must be purchased for a continued use.

Issues on Web site creations & Maintenance:-

A quality web site is *never* “finished??it is always a work in progress. Your visitors expect what they see to be fresh, with constant updates and improvements. Repeat visitors do not want to see exactly the same static content when they return weeks or months later.

To remain timely, a site requires regular maintenance and care. Some examples of key maintenance required might include:

- Editorial updates and revisions to keep information fresh and current.
- Changes to business practices, products offered, procedures and policies.
- Announcements, newsletters and helpful hints to support your customers and attract new business.
- Advice from an expert *(YOU!)* on how to respond or adapt to *today's* special challenges.
- “Special Offers?and time-sensitive information.
- Seasonal greetings and promotions.
- Re-design of the site “look?and “feel?to keep it fresh.
- Changing images to avoid being stale or static.
- Calendar of events (at least 2-3 months into the future).
- Additional features to make the site more interesting, interactive, and give it additional value.
- Additional submission or re-submission to Search Engines
- Ongoing review of Search Engine Position or ranking.
- Changes to staff or department contact information.
- Analysis of site usage including
    - Volume of visitors, and how long do they stay?

- Referral sources
- Most popular pages
- Trends by time period
- Types of browsers used
- Visitor location by country

We can design a Maintenance program based on the size of your site, the market you serve, the realities of your budget, and the dynamics that drive your business. Your company is unique, and your web presence should help to identify what sets you apart from the rest of the pack.

You determine the nature and frequency of updates. *You* determine when information becomes stale or stagnant and needs to be refreshed.

FTP software for upload web site:-

FTP or File Transfer Protocol is an internet protocol used to transfer files across the internet from one computer to another. For example, when you are installing a software like WordPress on your website, then you will most likely use FTP in order to upload all of the WordPress files from your personal computer to your server. In order to do this, you need something called an FTP client. The FTP client is software that runs on your personal computer and allows you to transfer files to and from your web server. Filezilla and WS_FTP are examples of easy to use FTP clients.

Most WordPress web hosting service providers give users access to FTP, so that they can upload or download files from their web hosting server directly from within their control panel. WordPress users may need an FTP client to upload WordPress files to their web hosting server before they can install WordPress.

Besides installing WordPress, if you plan on using plugins or modifying your theme’s functions.php file then it is good to understand how to use FTP. Sometimes adding a plugin or making a change to the functions.php file can cause the “white screen of death” and lock you out of your site due to poorly written code or compatibility issues between plugins. If you know how to use FTP, then you can simply connect to your server using your FTP client and delete the files that are causing the problems.

How to Upload a File to your Website using the FileZilla FTP Client:

The process of transferring a file from your computer to your website is often referred to as "uploading" that file or "publishing" it. For web hosts that support FTP (short for "File Transfer Protocol") or SFTP (a secure form of the File Transfer Protocol), you need a program called an "FTP client" to transfer the file. You can find a list of free FTP clients at https://www.thefreecountry.com/webmaster/freeftpclients.shtml

This tutorial deals with how you can transfer a file to your web server using a free FTP client known as FileZilla. Versions for Windows, Linux and Mac OS X are available. I will describe the Windows version of this program, but if you use another operating system, chances are that it will work very similarly.

Download and Install FileZilla

First, go to the FileZilla download page and obtain the appropriate version for your system. For Windows, get the Setup version; at the time I wrote this, it's the one labelled "(recommended)" under the Windows section.

Once you have downloaded the program, you will have to install it. Go to your desktop and doubleclick the file that you have just downloaded. Follow the instructions to install it to your hard disk.

Preliminary Steps

Before you can upload any file to your site, you will also need some information from your web host. In particular, you will need to find out the following:

- The name of the FTP server for your website. For example, your host may tell you that your FTP hostname is "ftp.example.com".
- Your user id or login username for your FTP account.
- Your password for your FTP account.
- The directory where you need to place your files so that they can be seen by a web browser visiting your site. For example, your host may tell you to place the files in a subdirectory called "www" or "public_html" or even the default directory that you see when you log into your FTP site.

- If your web host tells you that you do not have FTP access, you cannot use this tutorial. You will have to use whatever method the web host has designated for you to upload your files.

Steps to Uploading or Publishing a File to Your Web Server

1. For the purpose of this tutorial, I will assume that you wish to upload a file called "feedback.php". Everytime you see "feedback.php" mentioned, you can substitute that name with the name of the file you actually wish to upload. FileZilla does not restrict you to uploading only files of that name. You can upload images (eg, GIF, JPG, PNG, etc), HTML files, video clips, music files (eg MP3 files, WAV files, MIDI files), Perl scripts, PHP scripts, entire directories (ie, folders) containing files, and so on. For the curious: I use "feedback.php" as the example file because this guide was originally written to help those who use my free Feedback Form Wizard to upload the generated form to their website.
2. If you see a dialog box with the title "Site Manager" when you start up FileZilla, go to the next step. If not, click the "File" menu followed by the "Site Manager" item on that menu. A dialog box will appear.
3. Click the "New Site" button. This creates a new item under "My Sites" (or "My FTP Sites" depending on which version of FileZilla you are using) called "New site" (or "New FTP site" in older versions)
4. Rename "New site" (or whatever the initial name was) to the name of your site. By default, the keyboard cursor would have been placed in the name portion of "New site" allowing you to change it immediately. If you have lost the cursor because you accidentally clicked somewhere else in the dialog box, you can get it back by simply clicking once on the name. Note that this name can be anything you wish. It is not required for accessing your site. However, you will probably make your life easier if you change it to your site's name rather than some random string of characters.
5. On the right side of the dialog box, under the tab "General", enter the name of your FTP server in the "Host" input box. For example, if your web host told you that your FTP hostname is "ftp.example.com", enter "ftp.example.com" into the space provided.
6. Leave the "Port" entry box alone. If your web host tells you to connect using SFTP, click the drop-down box for the "Protocol" field, and select "SFTP - SSH File

Transfer Protocol". If they tell you to connect with FTP, choose "FTP - File Transfer Protocol". If they allow both, select the SFTP entry, because that will encrypt your connection, hopefully protecting your password from being intercepted by others.

7. Use your mouse and select "Normal" from the drop down list box for "Logon Type". This will enable the "User" and "Password" boxes for the next step.
8. Enter your user id or your login name (or whatever your web host calls it) into the "User" input box, and your password into the "Password" input box. Note that this information is automatically saved onto your computer and will be re-used the next time you run FileZilla, so you do not have to re-enter them again. (It also means that you should not use FileZilla in this way on computers that others may have access to, such as those found in an Internet cafe or a public library.)
9. Click the "Connect" button. FileZilla will proceed to log into your server. If it is successful, you will see a directory listing of your website's account on the right hand side of the FileZilla window. I shall refer to this as the Remote Site pane. The left hand side shows the directories and files on your own computer. This is the Locate Site pane.
10. If your web host told you to only upload in a specific subdirectory on the web server, such as in the "www" or "public_html" directory, change to that directory by doubleclicking its name in the folder portion of Remote Site pane. FileZilla will open that folder accordingly and show you its contents.
11. Next, locate the file that you wish to upload in the Local Site pane. Both window panes behave mostly like Windows Explorer windows, so navigating them should not be unduly hard. Once you have located the file you wish to upload, say "feedback.php", drag that file to the Remote Site pane. Another way is to right-click the file (that is, click the right mouse button while the pointer is hovering over the filename) and select "Upload".
12. The file will then be transferred to the folder that is currently open on the Remote Site pane, so make sure you have changed to the appropriate directory before initiating the upload.
13. If a file of the same name already exists on your web host's computer, FileZilla will pop up a dialog box asking you what to do. Click the checkbox "Apply to current queue only" to put a tick in it, followed by the "OK" button, and the program will

proceed to overwrite the existing file. On the other hand, if you realised that you have made a mistake and tried to transfer the wrong file, click the "Cancel" button to abort the transfer.

Elements of HTML syntax:-

An **HTML element** is an individual component of an HTML document or web page, once this has been parsed into the Document Object Model. HTML is composed of a tree of HTML nodes, such as text nodes. Each node can have HTML attributes specified. Nodes can also have content, including other nodes and text. Many HTML nodes represent semantics, or meaning. For example, the `title` node represents the title of the document.

HTML

(Hypertext Markup Language)

The official logo of the last version – HTML5.[1]

Filename extension	`.html`
	`.htm`

Internet media type	`text/html`
Type code	TEXT
Developed by	W3C & WHATWG
Initial release	1993; 25 years ago
Latest release	5.2[2] / 5.3 (working draft)[3] (14 December 2017; 5 months ago)
Type of format	Document file format
Extended from	SGML

Extended to	XHTML
Standards	ISO/IEC 15445 W3C HTML latest recommendation HTML Living Standard
Open format?	Yes
Website	www.w3.org/html/ whatwg.org

HTML

HTML

Dynamic HTML

HTML5

audio

canvas

video

XHTML

Basic

Mobile Profile

C-HTML

HTML element

span and div

HTML attribute

HTML frame

HTML editor

Character encodings

Unicode

Language code

Document Object Model

Browser Object Model

Style sheets

CSS

Font family

Web colors

HTML scripting

JavaScript

WebGL

WebCL

W3C

Validator

WHATWG

Quirks mode

Web storage

Rendering engine

Comparisons

Document markup languages

HTML support

XHTML

1.1

v

t

e

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.[4]

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

HTML Tags

HTML tags are the hidden keywords within a web page that define how your web browser must format and display the content.

Most tags must have two parts, an opening and a closing part. For example, <html> is the opening tag and </html> is the closing tag. Note that the closing tag has the same text as the opening tag, but has an additional forward-slash (/) character. I tend to interperet this as the "end" or "close" character.

There are some tags that are an exception to this rule, and where a closing tag is not required. The <img> tag for showing images is one example of this.

Each HTML file must have the essential tags for it to be valid, so that web browsers can understand it and display it correctly.

The rest of the HTML file can contain as little or as many tags as you want to display your content.

Tag Attributes

Attributes allow you to customise a tag, and are defined within the opening tag, for example: <img src="image1.jpg"> or <p align="center"> ... </p>

Attributes are often assigned a value using the equals sign, such as border="0" or width="50%", but there are some that only need to be declared in the tag like this: <hr noshade>.

Most attributes are optional for most tags, and are only used when you want to change something about the default way a tag is displayed by the browser. However, some tags such as the <img> tag has required attributes such as src and alt which are needed in order for the browser to display the web page properly.

Example:

Below is a basic html document, containing all the essential tags. You can copy the code below, paste it into your editor, and save as mypage.html to start your own web page.

```
<html>

 <head>
```

```
  <title>My Page Title</title>

 </head>

 <body>


  This is where all my web page content goes!


 </body>

</html>
```